УДК 004.056.5 Вестник УрФО № 3(21) / 2016, с. 30–37

Мищенко Е. Ю., Соколов А. Н.

КОЛИЧЕСТВЕННЫЙ АНАЛИЗ ПРОЦЕДУРЫ ОБЕЗЛИЧИВАНИЯ ПЕРСОНАЛЬНЫХ ДАННЫХ. МЕТОД ПЕРЕМЕШИВАНИЯ

Обезличивание – способ обработки персональных данных, целью которого является приведение этих данных в защищенное состояние, которое не позволяет злоумышленнику использовать их во вред физическому лицу. Результат обезличивания персональных данных зависит от их содержания и применяемого метода обезличивания. Нормативные акты определяют несколько методов обезличивания, но все они описываются качественными критериями. В статье производится количественный анализ одного из методов обезличивания – метода перемешивания. Предлагается вариант технической реализации данного метода, включая решение проблемы необходимого и достаточного подмножества перемешиваемых записей, определение требований к правилам перемешивания, а также рассмотрение возможных способов реализации таких требований. На основе реального примера производится оценка эффективности метода по различным критериям. В том числе по техническим критериям (невозможность идентификации, с одной стороны, и возможность деобезличивания с применением заданных правил, с другой стороны), а также по экономическим критериям (окупаемость). На базе показателей вероятности идентификации и степени обезличивания персональных данных приводятся рекомендации по повышению эффективности данного метода обезличивания персональных данных.

Ключевые слова: *персональные данные, обезличивание персональных данных, метод перемешивания.*

Mishchenko E. Yu., Sokolov A. N.

QUANTITATIVE ANALYSIS OF THE DEPERSONALIZATION PROCEDURE. METHOD OF MIXING

Depersonalization is the way of personal data processing for the purpose of transforming data to protected status, in order to prevent disturber use it to damage the person. The result of depersonalization is depending on content of the personal data and the depersonalization method also. Standard acts define some methods of depersonalization, but all of them are describing by qualitative criterions. This article makes the quantitative analysis of one of the de-

personalization methods – the method of mixing (MIX Method). Proposed variant of technical realization of MIX Method solves the problem of the size of mixing subarea, which is necessary and sufficient for the modification, define the requirements for mixing rules, and also describes the methods to realize them. On practical example the performance evaluation of MIX Method is made by using some criterions, including the technical criterions (identification impossibility on the one side and reconstruction possibility by means of modification rules on the other side), and commercial criterions (economic return). On the base of identification probability and depersonalization degree some recommendations of efficiency enhancement are proposed.

Keywords: *personal data, depersonalization, method of mixing.*

Приказом Роскомнадзора[1] предусмотрено четыре метода обезличивания, два из которых (метод введения идентификаторов и метод изменения состава или семантики) были рассмотрены с точки зрения количественного анализа в предыдущих статьях ([2] и [3]). В данной статье рассматривается метод перемешивания, который реализует принцип подстановки чужих значений (из другой записи этой же БД) вместо значений атрибутов обезличиваемой БД. Дополнительными данными, необходимыми для деобезличивания, являются параметры перестановки (перемешивания) атрибутов различных записей между этими записями.

Реализация метода перемешивания, как и метода изменения состава и семантики принципиально отличается от метода введения идентификаторов тем, что требует гораздо больше усилий в процессе перевода исходной базы в обезличенное состояние. Дело не только в том, что метод введения идентификаторов не предусматривает изменение атрибутов, но и в том, что в подавляющем большинстве существующих баз персональных данных используется реляционный принцип обработки данных, то есть база уже разделена на множество таблиц, связанных реальными (имеющими смысл) идентификаторами, и для реализации метода введения идентификаторов часть пути уже пройдена, остаются завершающие этапы. Если же в основе метода обезличивания лежит модификация атрибутов, то эти модифицируемые атрибуты уже гораздо сложнее использовать для связи между таблицами, поэтому придется либо частично отказываться от реляционного принципа обработки (снова объединять таблицы), либо вводить множество абстрактных идентификаторов.

Кроме вышеуказанного, реализация метода перемешивания, по сравнению с методом изменения состава и семантики, осложняется тем, что может охватывать не одни только идентифицирующие атрибуты, а абсолютно все атрибуты исходной базы ПД.

1. Описание метода. В соответствии с Приказом Роскомнадзора[1] метод перемешивания реализуется путем перемешивания отдельных записей, а так же групп записей между собой, и создания некоторого правила перемещения этих сведений. Перемещение должно быть однозначным (обратимым) для обеспечения деобезличивания, поэтому мы будем учитывать только способы перемещения «один в один». После применения данного метода единая база (БД), в отличие от метода введения идентификаторов, не распадается на части, а в отличие от метода изменения состава и семантики, обязательно сохраняет структуру и смысл атрибутов. В итоге возникают следующие особенности:

1) БД преобразуется в обезличенную базу с тем же объемом и идентичной структурой, в которой каждый отдельно взятый атрибут одного физического лица (ФЛ) имеет прежнюю семантику, но получает значение из другой записи базы (второго ФЛ), причем другой атрибут первого ФЛ может получить значение аналогичного атрибута третьего ФЛ. При этом любой атрибут первого ФЛ может остаться неизменным. Таким образом, для злоумышленника конечная БД не выглядит ни обезличенной, ни даже модифицированной.

2) Создаются алгоритмы прямого и обратного перемещения значений атрибутов между записями. Алгоритмы должны быть формализованы для автоматизированного использования (с помощью ПО).

В данном процессе модификации БД необходимо решить три проблемы:

1) Какие атрибуты включить в группу перемешиваемых.

2) Каков объем группы перемешиваемых записей.

3) Какими свойствами должен обладать алгоритм перемешивания.

1.1. Атрибуты модифицируемой группы. Набор атрибутов, подлежащих модификации (набор перемешивания), в отличие от

методов введения идентификаторов или изменения состава и семантики, казалось бы, не должен соответствовать каким либо специальным требованиям. Единственное требование – неизвестность для злоумышленника (т.е. количество и состав таких атрибутов является секретом метода). Однако, некоторые нестрогие ограничения можно обозначить:

• набор перемешивания не должен быть слишком мал, иначе злоумышленник легко вычислит его на примере известного ему ФЛ (возможно, себя!) и сделает вывод о достоверности прочих атрибутов.

• набор перемешивания не обязательно должен включать все идентифицирующие атрибуты (смотри статьи [2] и [3]), но хотя бы часть их должна туда входить (интегральный показатель ВИ для этой части набора лишь в идеале должен быть равен 1, но можно допустить и меньшие значения).

• желательно, чтобы в набор перемешивания входили хотя бы некоторые не идентифицирующие атрибуты. Причина – та же, что и для первого ограничения.

• поскольку в основе метода лежит перемещение атрибутов в другие записи, то должен существовать один неперемещаемый атрибут, привязывающий данную запись не обезличенной базы к конкретному ФЛ – некий аналог номера записи. Поскольку этот атрибут должен быть уникальным, то он не может быть характеристикой ФЛ, т.е. должен быть абстрактным.

Достаточность и состав идентифицирующего набора атрибутов подробно рассматривалась в предыдущих статьях [2] и [3], поэтому включение в набор перемешивания всех таких атрибутов настоятельно рекомендуется.

Что касается всех остальных атрибутов, то в статье[2] оценивается вероятность идентификации ФЛ по таким реквизитам (ВИ зависит от атрибутов, но в общем случае близка к 0,0001), но если эти атрибуты не трогать совсем, то возникает опасность вычисления алгоритма перемешивания. Следовательно, в идеальном варианте в набор перемешивания надо включить все атрибуты БД, что несколько замедлит процедуру обработки, зато избавит от необходимости обосновывать не включение некоторых из них.

В статье[3] уже обращалось внимание на необходимость исключения из модифицируемого набора специальных идентификаторов (искусственно введенных атрибутов), которые связывают различные таблицы базы данных реляционного типа (справочники с одной стороны и изменяемые данные функционального характера – с другой). Однако, для метода перемешивания это не является критичным, атрибуты этого типа могут включаться в модифицируемый набор, тем более если эти идентификаторы являются идентифицирующими ФЛ атрибутами.

1.2. Требования к объему группы перемешивания. В процессе перемешивания каждого конкретного атрибута участвует минимум (в простейшем варианте - обмене) две записи ФЛ: значение атрибута из одной записи сохраняется в буфере обмена, на его место записывается значение этого же атрибута из второй записи, а во вторую запись заносится значение из буфера. В данном случае эти две записи и составят группу перемешивания. Поскольку перемешиванию подвергается несколько атрибутов (возможно – все), то в обмене каждого следующего атрибута должна участвовать уже другая (ранее не участвовавшая в обмене) запись. То есть для обмена всех атрибутов потребуется группа перемешивания, по количеству записей равная количеству атрибутов. Хотя это требование нельзя назвать строгим (если, например, атрибутов всего 30, а записей – 29, т.е. не перемешанным оказался один атрибут, вряд ли ВИ будет сильно отличаться), целесообразнее его выполнить, чем рассчитывать ВИ для других вариантов. Таким образом, значения 30-ти атрибутов одной записи будут содержаться в 30-ти других записях группы перемешивания, а в самой этой записи будут содержаться значения из упомянутых 30-ти разных записей. Но это совсем не означает, что группа перемешивания будет ограничена только 30 записями. Ведь из второй записи группы в первую запись попадет только один атрибут, а остальные атрибуты могут быть записаны в 31-ю, 32-ю и т.д. записи, т.е. группа перемешивания увеличится на 29 записей, и то же самое произойдет при перемешивании следующей записи. Следуя данному алгоритму, мы очень быстро включим в группу перемешивания все записи БД, а сама задача решена не будет, поскольку группа не будет замкнутой. Но данное противоречие говорит лишь о том, что был выбран неправильный алгоритм, а размер группы перемешивания является важнейшим параметром алгоритма.

Требование замкнутости группы перемешивания приводит к необходимости дискретных значений количества записей в группе –

оно должно быть кратно количеству перемешиваемых атрибутов. Хотя это не означает, что размер группы должен быть постоянным. Для нашего примера с 30 атрибутами внутри одного алгоритма допустимы группы по 30, 60, 90 записей. Однако при этом возникает проблема некратной последней группы из-за некратного и постоянно меняющегося общего количества записей в БД, но эту проблему изменением размера группы не решить.

1.3. Требования к алгоритму перемешивания. Аналогично методу изменения состава и семантики алгоритм является секретом также для метода перемешивания, и суть этого секрета составляют параметры перемешивания. Два таких параметра мы уже рассмотрели: количество атрибутов и количество записей, принимающих участие в некотором заданном этапе – назовем его циклом перемешивания. Третьим параметром перемешивания является алгоритм перемещения значений в цикле перемешивания. Алгоритм перемещения может быть задан как в виде таблицы подстановок (соответствие «номер записи – номер записи») для каждого атрибута, так и в виде формулы, связывающей номер исходной записи, номер атрибута в этой записи и номер конечной записи в процессе перемещения. Формула может быть любой сложности, но существует достаточно жесткое ограничение: однозначное соответствие. Однозначность соответствия означает, что полный охват заменяемых атрибутов в заданной группе перемешивания должен сопровождаться полным охватом замененных атрибутов в этой же группе, причем акт замены каждого атрибута должен быть однократным в одном цикле перемешивания (все атрибуты группы перемешивания заменяются за один раз).

Количество циклов перемешивания для конкретной группы перемешивания не обязательно ограничивать одним циклом. Назовем совокупность циклов над одной группой перемешивания циклом первого порядка. Кроме того, возможны циклы второго и более высоких порядков – когда после отработки циклов первого порядка над всеми группами перемешивания можно задать циклы обмена данными между различными группами. После охвата циклами второго порядка всех групп процедуру можно повторить (порядок циклов увеличится).

Обязательным условием для любого алгоритма является решение проблемы некратной последней группы перемешивания. Можно предложить дополнить ее пустыми (шумовыми) записями, а можно – дублями любых других записей. Понятно, что лишние записи должны быть специально помечены.

2. Оценка эффективности метода. В предыдущих статьях [2] и [3] при оценке эффективности методов введения идентификаторов и изменения состава и семантики использовался один и тот же подход, рассчитанный на то, что в обезличенной базе все не модифицированные атрибуты однозначно принадлежат одному (идентифицируемому) ФЛ. Именно этот факт играет решающую роль как при попытках идентификации ФЛ злоумышленником, так и при обработке обезличенных данных оператором. Кроме того, этот факт определяет тесную зависимость интегральной ВИ от семантики функциональных (не идентифицирующих) атрибутов (т.е. при одном и том же методе введения идентификаторов для базы регистрации автомобилей и для базы посещения поликлиники интегральная ВИ по всем функциональным атрибутам будет отличаться, хотя ВИ для отдельных аналогичных реквизитов может совпадать).

Метод изменения состава и семантики по критерию анонимности оказался в целом (без применения шифрования) слабее метода введения идентификаторов, поскольку последний полностью исключает возможность вычисления (подбора) алгоритма.

В методе перемешивания атрибуты одной записи базы принадлежат разным ФЛ, что делает невозможным применение метода оценки эффективности, использованного для предыдущих методов обезличивания. Однако, принципиальное наличие идентифицирующих атрибутов в обезличенной базе, аналогично методу состава и семантики, не исключает возможности вычисления (подбора) алгоритма перемешивания.

2.1. Эффективность анонимности. В предыдущих статьях [2] и [3] мы производили оценку значения СО (степень обезличивания, определяемая по формуле СО = 1 - ВИмакс) на примере БД регистрации автомобилей (злоумышленник искал идентифицирующие атрибуты ФЛ на основании функциональных атрибутов его автомобиля при условии свободного доступа к обезличенной базе). И хотя для метода перемешивания семантика базы в целом не имеет значения (семантика отдельного атрибута – имеет!), мы продолжим в качестве примера рассматривать данную базу, обезличенную методом перемешивания.

Из открытых источников злоумышленник может получить и использовать следующую информацию:

1) Злоумышленнику известен регион, в котором ФЛ эксплуатирует свой автомобиль (средний регион нашей страны - с населением 2 млн. человек, областной центр - с населением 1 млн. человек);

2) Возраст ФЛ – от 18 до 60 лет;

3) Количество ДТП в год по региону – 3 тыс. при количестве автомобилей – 800 тыс., по областному центру – 2 тыс. при количестве автомобилей – 500 тыс.

4) Для оценки максимального значения ВИмакс, примем критерий «актуальность идентификации» (т.е. время, доступное злоумышленнику на поиск) равным 30 дням.

В предыдущих статьях [2] и [3] мы уже получили оценку ВИмакс=1/7500 для областного центра без использования атрибутов из БД. При использовании атрибутов из обезличенной базы при применении метода введения идентификаторов ВИмакс увеличивалась до 1/1500 (СО=1-1/1500=0,9993). При использовании метода изменения состава и семантики ВИмакс повышалась, но крайне незначительно, в результате чего СО могла понизиться в пределах 0,0001. Рассмотрим, как поведет себя ВИ в методе перемешивания.

Итак, злоумышленник имеет ВИмакс=1/7500 (0,00013). Если он обнаружит (т.е. свяжет друг с другом) в перемешанной базе все известные ему функциональные атрибуты («марка», «модель», «цвет кузова», «государственный номер»), он все равно не получит ВИмакс=1/1500, не вычислив неизвестные ему функциональные атрибуты («дата регистрации», реквизиты полиса ОСАГО и т.п), не говоря уже о более высоких значениях ВИ. Более того, поскольку даже известные ему четыре атрибута разбросаны по разным записям, вероятность их точного вычисления будет меньше 1, что в результате уменьшит ВИмакс, а не увеличит. Для нашего примера примем количество атрибутов равным 30. Вероятность принадлежности заранее известного значения атрибута искомому ФЛ определяется весом значения атрибута. Задача злоумышленника облегчается в случае, если среди известных ему функциональных атрибутов есть уникальные. В нашем примере это «государственный номер» - для него ВИ=1, т.е. первый шаг к увеличению ВИмакс можно считать полностью успешным, как только этот атрибут найден в базе. Дополнительно гарантируется, что в этой записи других атрибутов искомого ФЛ нет. Отметим, что задача злоумышленника облегчается тем, что модель автомобиля почти полностью коррелирует с маркой (исключения редки, и мы ими пренебрежем), т.е. в качестве следующего шага он будет искать именно модель. Для нашего примера выберем среднюю по стоимости и достаточно «молодую» (чтобы не учитывать более двух прошлых лет) модель – «Шкода Рапид». Всего в стране их около 17000, в нашем городе-миллионнике автомобилей этой модели – около 400. Значит, при нахождении данного значения модели в базе есть вероятность 1/400 (0,0025), что в этой записи находится значение для искомого автомобиля, и нет других атрибутов, относящихся к его владельцу. Последний известный атрибут «цвет» имеет для данной модели 7 различных значений. Но поскольку аналогичный цвет может быть и у других моделей, то вероятность обнаружить искомое значение будет не 1/7, а гораздо меньше – вплоть до 1/50000 (белый и черный цвет есть у всех моделей). Это и есть вероятность того, что будут правильно связаны все три атрибута, а это уже меньше исходного ВИмакс. Значит, такой метод поиска (идентификации) неэффективен против метода перемешивания. Однако, есть другой метод, который является более эффективным.

Этот метод можно назвать «шаблоном», и состоит он в следующем. Обязательным условием идентификации по шаблону является гарантированное наличие в обезличенной базе заранее известной информации хотя бы об одном ФЛ (лучше – о нескольких). Отметим, что «шаблон» не применим для метода введения идентификаторов, а также для метода изменения семантики. Но в случае изменения состава (структуры базы) он может быть эффективным.

В нашем примере это может быть информация об автомобиле самого злоумышленника (идентификация самого себя). Обнаружив все свои реквизиты, можно составить таблицу подстановки «атрибут» - «номер записи» и далее, перемещая полученный шаблон по базе, вычислить реальные значения атрибутов для любого ФЛ. Первым шагом будет поиск в базе государственного номера своего автомобиля, который даст ВИ=1. В качестве второго шага можно было бы выбрать «адрес проживания», но скорее всего, он разбит на части (с использованием справочника улиц),

поэтому выберем атрибут «номер мобильного телефона» - ВИ=1, и т.д. по мере уменьшения уникальности. Для атрибута «фамилия» ВИ будет от 1 до 1/100. Облегчим задачу злоумышленнику и примем ВИ=1/10. Кроме того, учтем, что количество водителей (по возрастному признаку) в два раза меньше количества жителей. Для даты рождения ВИ=1/25, для имени – от 1/14500 (Александр) до 1, можно принять ВИ=1/300 (Альберт). Для модели автомобиля мы уже ВИ определили (1/400), остальные атрибуты будут давать все меньшие значения ВИ, поэтому их рассматривать пока не будем. Однако даже для 5 самых надежных реквизитов (при двух уникальных!) вероятность идентификации (построения шаблона) для известного ФЛ в перемешанной базе меньше 1/75000, что мало отличается от прямого перебора и явно менее эффективно, чем пытаться найти искомый автомобиль на стоянке у наиболее вероятного пункта оформления вероятного ДТП, не пользуясь базой вообще.

Но если применен не очень сложный алгоритм перемешивания (например, группа перемешивания включает 30 близко расположенных записей, и используется один цикл перемешивания), то после нахождения «государственного номера» злоумышленник сможет локализовать область поиска – она будет иметь удвоенный размер группы перемешивания, точнее, 59 записей. При этом ВИ для всех атрибутов значительно увеличатся. Например, вероятность повтора фамилии в такой группе равна 0,003, модели автомобиля – 0,048, улицы (как части адреса проживания) – 0,12. Но поскольку вероятность повтора двух атрибутов меньше 0,01, один повторный атрибут можно легко вычислить методом исключения. Убедиться в применении такого алгоритма тоже просто. Если второй уникальный атрибут («номер мобильного телефона») расположен в пределах 30 записей от первого, то, скорее всего, остальные атрибуты тоже близко. Кстати, область поиска при этом уменьшится (найденные атрибуты могут оказаться на границах группы перемешивания).

Таким образом, в случае использования упрощенного алгоритма злоумышленник легко построит шаблон и вычислит атрибуты любого ФЛ. Но при увеличении группы перемешивания до 1200 записей повторы (и неоднократные – 3 одинаковых значения для улицы и 3 для номера дома!) одновременно нескольких реквизитов станут неизбежными, и злоумышленник получит более 50 разных шаблонов и вряд ли успеет их проверить за 30 дней. Однако, для надежности лучше увеличить объем группы перемешивания до 10000 записей, хотя более эффективным было бы усложнить алгоритм (например, заложить несколько разных шаблонов для различных групп перемешивания или увеличить количество циклов перемешивания – в этом случае ВИ уменьшится в соответствующее количество раз).

2.2. Эффективность применимости. Для оценки применимости рассматриваемого метода необходимо убедиться в наличии технической возможности реализации всех этапов технологического процесса обработки информации (ввод, изменение, хранение, передача, вывод) – как в обезличенном, так и в восстановленном виде. А уже при наличии технической возможности определяющую роль играют стоимость и сроки реализации.

На рис.1 приведена схема разделения обработки ПД на защищаемых и не защищаемых рабочих местах, где в качестве связующего звена используется межсетевой экран.

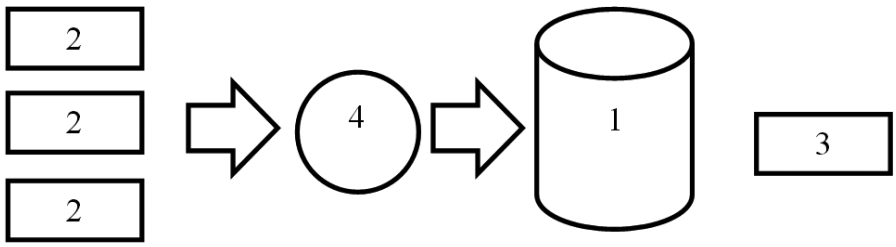

Рис. 1. Схема разделения обработки персональных данных на защищаемых и не защищаемых рабочих местах

Цифрами на рис.1 обозначены:

1 – Обезличенные данные (сервер обезличенной базы, в свободном доступе, не защищается);

2 – Рабочее место оператора ПД с установленным алгоритмом перемешивания (входит в состав ИСПДн, защищается);

3 – Рабочее место оператора обезличенной базы (в свободном доступе, не защищается);

4 – Межсетевой экран, обеспечивает односторонний доступ к информации, направление которого указано стрелками.

В нашем примере с БД регистрации автомобилей перемешиванию подвергаются не только чувствительные к идентификации ПД владельцев (фамилия, имя, отчество, дата рождения, место рождения, адрес проживания, телефон, номер паспорта), но и открытые данные о постановке и снятии с учета всех автомобилей. Перемешивание и восстановление производится только пользователями рабочего места 2 (регистратура). А в базе 1 будут храниться и обрабатываться все атрибуты уже в перемешанном виде, и доступ к ним будет разрешен не только легальным пользователям рабочих мест 3, но и любым пользователям, которые смогут (в принципе без ограничений) получить физический доступ к базе 1 (например, подключившись к базе 1 по Интернет).

Анонимность обеспечивается межсетевым экраном, который запрещает доступ к информации (ПД) и алгоритму перемешивания на рабочем месте 2 от рабочих мест на стороне обезличенной базы, то есть злоумышленник (как и любой пользователь рабочего места 3) не имеет доступа к алгоритму.

Для определения эффекта применимости показанной на рис.1 системы в целом необходимо оценить применимость ее обеих частей – защищаемой (состоит из необходимого количества рабочих мест 2) и не защищаемой (состоит из сервера 1 и некоторого количества рабочих мест 3).

Пользователь рабочего места 2 может решать следующие задачи:

1) указав значение любого идентификатора конкретного ФЛ, выдать запрос через экран 4, получить предварительно данные нескольких ФЛ (включая неперемещаемые «номера записей») из обезличенной базы (сервер 1) и сопоставить их с конкретным ФЛ (и подтвердить его выбор);

2) указав значения открытых атрибутов из обезличенной базы 1, получить список из нескольких ФЛ с заданными значениями атрибутов и восстановить их в виде ПД;

3) только этот пользователь может добавлять записи о новых ФЛ (создавая новые «номера записей», а также удалять ФЛ из БД полностью (т.е. всю информацию о ФЛ).

Необходимо отметить, что как добавление, так и удаление записей является очень сложной, трудоемкой операцией с далеко не очевидным результатом. Например, при добавлении необходимо разместить реквизиты нового ФЛ в несколько других записей из состава последней неполной группы перемешивания, а она может быть уже заполнена. А при удалении ФЛ из базы в ней появится множество «дырок», о заполнении которых необходимо периодически заботиться (в алгоритме должна быть предусмотрена «дефрагментация базы»)

Таким образом, применимость (техническая реализация функций) с точки зрения защищаемой системы хоть и сложнее, чем в других методах, тем не менее вполне очевидна.

Пользователь рабочего места 3 может выполнять аналогичные рабочему месту 2 функции 1) и 2), но в ограниченном варианте, так как не может сопоставить эту информацию конкретным ФЛ, причем обе эти функции не отличаются друг от друга с точки зрения результата и имеют только статистическое значение (количество ФЛ, суммарные и средние величины и т.д.), а функцию 3) он выполнять не может в принципе.

Что касается функции ввода существенной информации (ради которой и создана вся система обработки), то в отличие от методов введения идентификаторов или изменения состава и семантики, она может выполняться только пользователем 2, а пользователем 3 не может принципиально. Очевидно, что применимость рабочего места 3 крайне ограничена и вызывает серьезные сомнения.

Аналогично рассмотренным ранее реализациям методов обезличивания при первом посещении (постановка автомобиля на учет) владелец придет к рабочему месту 2, представит свои ПД (паспорт) и получит идентификатор, но в нашем случае ему достаточно дать абстрактный технологический идентификатор, который является основным параметром алгоритма – не перемещаемый «номер записи». Напомним, что по методу введения идентификаторов это был идентификатор связи с таблицей соответствия, а по методу изменения состава и семантики идентификатор похож на «номер записи», но его значение в алгоритме модификации никакой роли не играет. После получения идентификатора ФЛ-владелец, в отличие от других методов, не идет к рабочему месту 3 (это бессмысленно), а представляет все данные об автомобиле на рабочем месте 2. При повторном посещении (снятие с учета или постановка на учет другого автомобиля) владелец идет с идентификатором снова к рабочему месту 2. Тип внешнего носителя, на котором хранится идентификатор ФЛ, и

проблемы, связанные с его применением, уже обсуждались в статье[3].

Данный метод обезличивания имеет резко отличную от других методов эффективность применения, хотя некоторые сходные черты с методом изменения состава и семантики можно обнаружить:

1) Количество мест стационарного хранения «секрета метода» в обоих методах – количество рабочих мест 2 с установленным алгоритмом и установочных носителей (могут быть внешние носители с ключами алгоритма), кроме этих рабочих мест алгоритм известен разработчику;

2) Частичная утечка «секрета метода». В обоих методах любая утечка алгоритма – полная утечка всей базы до смены алгоритма, но возможна и частичная утечка при использовании одного «шаблона», в случае, если их несколько;

3) Доступность базы. В обоих методах БД всегда в целом виде;

4) Передача по каналам связи – приоритетный режим применения обоих методов.

2.3. Экономическая эффективность. Аналогично другим методам обезличивания, в нашем случае наибольшие финансовые затраты также потребуются на модернизацию программного обеспечения (ПО) ИСПДн. И это будет серьезным препятствием, когда производителем ПО является сторонняя организация.

Поскольку создание системы защиты ИСПДн требуется для всех рабочих мест, экономия может получиться на применении сертифицированных средств шифрования при передаче ПД между подразделениями. Для больших ИСПДн стоимость таких средств не зависит от количества рабочих мест, и для каждого подразделения составляет порядка 300000 руб.

Если сравнить эти затраты со стоимостью модернизации структуры БД и ПО (также не зависит от количества рабочих мест и составляет от 0 (ПО собственного производства) до 300000 руб.), то целесообразность обезличивания при наличии нескольких подразделений будет однозначной.

Таким образом, рассмотренный метод перемешивания характеризуется большей сложностью технической реализации, чем метод введения идентификаторов и даже метод изменения состава и семантики. В то же время эффективность обезличивания в зависимости от алгоритма модификации может быть гораздо выше, чем у метода введения идентификаторов, тем более, чем к метода изменения состава и семантики. К сожалению, применимость метода ограничивается либо передачей данных, либо их внешним хранением.

Примечания

1. Приказ Роскомнадзора от 5.09.2013 г. № 996 «Об утверждении требований и методов по обезличиванию персональных данных» [электронный ресурс]. URL: http://www.garant.ru

2. Мищенко Е.Ю., Соколов А.Н. Количественный анализ процедуры обезличивания персональных данных. Метод введения идентификаторов // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2015. Т. 15, № 3. С. 18–25.

3. Мищенко Е.Ю., Соколов А.Н. Количественный анализ процедуры обезличивания персональных данных. Метод изменения состава или семантики // Вестник УрФО. Безопасность в информационной сфере. — Челябинск: Изд. центр ЮУрГУ, 2016. № 1(19) С. 30–38.

Мищенко Евгений Юрьевич, старший преподаватель кафедры защиты информации, Южно-Уральский государственный университет, г. Челябинск. E-mail: Eug6303@mail.ru.

Соколов Александр Николаевич, канд. техн. наук, доцент, заведующий кафедрой защиты информации, Южно-Уральский государственный университет, г. Челябинск. E-mail: ANSokolov@inbox.ru.

E.Yu. Mishchenko, South Ural State University, Chelyabinsk, Russian Federation. E-mail: Eug6303@mail.ru,

A.N. Sokolov, South Ural State University, Chelyabinsk, Russian Federation. E-mail: ANSokolov@inbox.ru