электронное научно-техническое издание
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС 77 - 30569. Государственная регистрация №0421100025. ISSN 1994-0408

Процедура обезличивания персональных данных

 # 03, март 2011
УДК: 681.3
авторы: Саксонов Е. А., Шередин Р. В.

Московский государственный институт электроники и математики (МИЭМ)
saksmiem@mail.ru

Введение.

Вступление в действие Федерального закона о персональных данных обусловило разработку различных подходов, связанных с выполнением требований к их защите и сокращением издержек на ее обеспечение [1].

Одним из эффективных подходов к защите персональных данных является их обезличивание, поскольку оно позволяет снизить требования к уровню защищенности данных и, соответственно, сократить расходы на защиту. Поэтому процедуры обезличивания достаточно широко применяются на практике [1, 2, 3].

Под обезличиванием персональных данных будем понимать действия, в результате которых невозможно определить принадлежность персональных данных конкретному субъекту персональных данных.

Описание проблемы.

К настоящему времени можно выделить, например, следующие методы обезличивания персональных данных [2]:

- уменьшение перечня обрабатываемых сведений;
- замена части сведений идентификатором/ами;
- замена численных значений минимальным, средним, или максимальным значением;
- понижение точности некоторых сведений;
- деление сведений на части и обработка в разных информационных системах.

Критерием качества метода обезличивания часто является возможность определить на основании имеющихся обезличенных данных конкретного человека, при учете контекста обработки, но часто возможно использование дополнительной информации из других источников, позволяющей провести де-обезличивание [5, 7].

Многие из перечисленные методов не гарантируют невозможность получения персональной информации (де-обезличивания) путем использования контекста обработки и данных, размещенных в других системах, которые можно связать с обезличенными, поскольку эти методы, как правило, сохраняют связь между различными данными, относящимися к одному и тому же субъекту.

Разорвать эту связь возможно, если осуществить перемешивание данных, относящихся к различным субъектам.

Перемешивание данных имеет ряд достоинств, которые делают этот подход к обезличиванию достаточно перспективным:

- данные находятся в одном хранилище;
- использование дополнительных сведений, получаемых из других источников, не позволяет провести процедуру де-обезличивания;
- простота реализации обезличивания и обратного формирования персональных данных;
- мобильность данных, позволяющая распространять их, хранить в распределенных системах.

Однако, практическая реализация процедур обезличивания, основанных на перемешивании данных, в условиях, когда число хранимых данных достигает 10^6 - 10^9, требует преодоления значительных сложностей, связанных с описанием и заданием параметров перемешивания, разработкой математического и программного обеспечения.

Здесь предлагается процедура обезличивания, основанная на перемешивании данных, позволяющая оперировать с большими объемами данных, при простом задании параметров и большое количество возможных вариантов, обеспечивающее высокую защищенность от проведения де-обезличивания.

Описание задачи.

Предлагаемая процедура обезличивания, основана на разбиении исходного множества данных на подмножества, что позволяет сократить размерность и упростить ее практическую реализацию.

В качестве базового алгоритма процедуры предлагается использовать циклические перестановки [4].

Пусть задана исходная таблица персональных данных $T(t_1, t_2, ..., t_N)$, где N число атрибутов, M – длина таблицы. Будем рассматривать множество данных, относящееся к

одному атрибуту - t_i (i = 1,2,...,N). Это множество атрибута t_i - U_i, содержит M элементов. Все элементы каждого множества U_i занумерованы от 1 до M, и в таблице $T(t_1,t_2,...,t_N)$ совокупность элементов множеств разных атрибутов с одинаковыми номерами будем называть записью с соответствующим номером. Считаем, что в исходной таблице каждая запись имеет определенный смысл, связанный с конкретным субъектом (физическим лицом), т.е. содержит персональные данные конкретного лица, определенного в этой же записи.

Ниже приводятся описание и результаты анализа процедуры обезличивания.

Описание процедуры обезличивания.

Процедура обеспечивает перемешивание данных каждого множества атрибутов исходной таблицы на двух уровнях. На каждом уровне используется алгоритм циклических перестановок.

Первый уровень. Проведем разбиение множества U_i на K_i ($M > K_i > 1$) непересекающихся подмножеств U_{ij}, где число элементов подмножества U_{ij} равно M_{ij} ($M > M_{ij} > 0$), $j = 1,2,...,K_i$. Все элементы каждого подмножества U_{ij} считаем занумерованными от 1 до M_{ij} эти номера будем называть внутренними номерами элементов подмножества. Внешний номер элемента в подмножестве U_{ij}, имеющего внутренний номер k, обозначим - m_{ijk}, ($1 \le m_{ijk} \le M$). Так, что m_{ijk} - это порядковый номер элемента в множестве U_i, соответствующий элементу с внутренним номером k.

Разбиение каждого множества должно обладать следующими свойствами:

1) $U_i = \bigcup_{j=1}^{K_i} U_{ij}$ - подмножества разбиения включают все элементы множества U_i;

2) $U_{ij} \neq \varnothing$ и $U_{ij} \bigcap U_{im} = \varnothing$ для всех $j,m = 1,2,...,K_i$ - каждое подмножество не пусто, а пересечение любых двух подмножеств - пусто;

3) $m_{ij1} = m_{i(j-1)M_{i(j-1)}} + 1$ для всех $j = 2,3,...,K_i$ - для любых двух подмножеств с U_{ij} и $U_{i(j-1)}$ элемент с первым внутренним номером подмножества U_{ij} имеет внешний номер на единицу больший, чем внешний номер элемента с наибольшим внутренним номером подмножества $U_{i(j-1)}$;

4) если $k_1 > k_2$, то $m_{ijk_1} > m_{ijk_2}$ для всех $i = 1,2,...,N;\ \ j = 1,2,...,K_i$ - упорядоченность внешней и внутренних нумераций для всех множеств и подмножеств их разбиения совпадают;

5) $M = \sum_{j=1}^{K_i} M_{ij}$ - суммарное число элементов всех подмножеств U_{ij} равно общему числу элементов множества U_i.

Для каждого подмножества U_{ij} определим циклическую перестановку (подстановку), $c_{ij}(r_{ij})$ задаваемую следующим образом [4]:

$$c_{ij}(r_{ij}) = \begin{pmatrix} 1 & 2 & 3 & ... & (M_{ij}-1) & M_{ij} \\ (M_{ij}-r_{ij}+1) & (M_{ij}-r_{ij}+2) & (M_{ij}-r_{ij}+3) & ... & (M_{ij}-r_{ij}-1) & (M_{ij}-r_{ij}) \end{pmatrix}.$$

Здесь элементы первой строки матрицы, стоящей в правой части равенства, соответствуют внутренним номерам элементов подмножества U_{ij} до перестановки (в исходной таблице), а элементы, стоящие во второй строке, соответствуют внутренним номерами элементов подмножества U_{ij}, стоящим на местах, с номерами, определенными в верхней строке, после перестановки.

Таким образом, в перестановке (подстановке) $c_{ij}(r_{ij})$ производится циклический сдвиг всех элементов подмножества на число r_{ij}, $(1 \le r_{ij} \le (M_{ij}-1))$. Будем называть величину r_{ij} параметром перестановки $c_{ij}(r_{ij})$. Теперь все перестановки для всех подмножеств множества U_i можно задать набором (вектором) параметров $\mathbf{r}_i = (r_{i1}, r_{i2}, ..., r_{iK_i})$. Вектор параметров перестановок $\mathbf{r}_i$ задает первый уровень алгоритма перемешивания, т.е. перестановки первого уровня.

Второй уровень. Рассмотрим теперь множество $u_i = (u_{i1}, u_{i2}, ..., u_{iK_i})$, состоящее из K_i элементов. Здесь элемент u_{ij} соответствует подмножеству U_{ij}, $j = 2,3,...,K_i$. Для этого множества определим циклическую перестановку $c_{0i}(r_{0i})$:

$$c_{0i}(r_{0i}) = \begin{pmatrix} 1 & 2 & 3 & ... & (K_i-1) & K_i \\ (K_i-r_{0i}+1) & (K_i-r_{0i}+2) & (K_i-r_{0i}+3) & ... & (K_i-r_{0i}-1) & (K_i-r_{0i}) \end{pmatrix},$$

где элементы верхней строки матрицы перестановки соответствуют исходным номерам

элементов множества u_i (подмножеств U_{ij}), а элементы нижней строки матрицы соответствуют номерам элементов множества u_i, стоящим на местах с номерами, определенными в верхней строке, после перестановки.

Таким образом, в перестановке $c_{0i}(r_{oi})$ производится циклический сдвиг элементов множества u_i (подмножеств множества U_i) на число r_{0i}, $(1 \le r_{0i} \le (K_i - 1))$ – параметр перестановки. Эту перестановку будем называть перестановкой второго уровня.

В результате последовательного проведения перестановок первого и второго уровней получается перемешивание элементов множества U_i так, что меняется нумерация этих элементов по отношению к исходной нумерации.

Определим теперь нумерацию элементов множества U_i после проведения всех перестановок. Имеем, с учетом правил перемножения перестановок, следующую результирующую перестановку [4]:

$$c_i(r_{0i},\mathbf{r}_i)=$$

$$=\begin{pmatrix} [1 \quad \dots \quad M_{i(K_i-r_{0i}+1)}] & [M_{i(K_i-r_{0i}+1)}+1 \quad \dots \quad (M_{i(K_i-r_{0i}+1)}+M_{i(K_i-r_{0i}+2)})] & \dots & [M-M_{i(K_i-r_{0i})} \quad \dots \quad M] \\ m_{i(K_i-r_{0i}+1)1} \quad \dots \quad m_{i(K_i-r_{0i}+1)M_{i(K_i-r_{0i}+1)}} & m_{i(K_i-r_{0i}+2)1} \quad \dots \quad m_{i(K_i-r_{0i}+2)M_{i(K_i-r_{0i}+2)}} & \dots & m_{i(K_i-r_{0i})1} \quad \dots \quad m_{i(K_i-r_{0i})M_{i(K_i-r_{0i}}} \end{pmatrix}$$

Здесь верхняя строка матрицы содержит порядковые номера элементов множества атрибута i, в соответствии с их размещением в столбце после перемешивания, а нижняя строка содержит внешние номера элементов множества этого атрибута, соответствующие их размещению в исходной таблице.

Пример 1. Пусть $M = 15$, $K_i = 4$ и $M_{i1} = 4, M_{i2} = 4, M_{i3} = 4, M_{i4} = 3$ при этом $t_i = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15})$ и $U_{i1} = (a_1, a_2, a_3, a_4), U_{i2} = (a_5, a_6, a_7, a_8), U_{i3} = (a_9, a_{10}, a_{11}, a_{12}), U_{i4} = (a_{13}, a_{14}, a_{15})$ и $\mathbf{r}_i = (2,1,2,1)$, $r_{0i} = 2$.

Имеем, после применения перестановок первого уровня:

$$c_{i1}(2)=\begin{pmatrix} 1 & 2 & 3 & 4 \\ a_3 & a_4 & a_1 & a_2 \end{pmatrix}, \quad c_{i2}(1)=\begin{pmatrix} 1 & 2 & 3 & 4 \\ a_6 & a_7 & a_8 & a_5 \end{pmatrix},$$

$$c_{i1}(2)=\begin{pmatrix} 1 & 2 & 3 & 4 \\ a_{11} & a_{12} & a_9 & a_{10} \end{pmatrix}, \quad c_{i4}(1)=\begin{pmatrix} 1 & 2 & 3 \\ a_{14} & a_{15} & a_{13} \end{pmatrix}.$$

Результирующая перестановка имеет вид:

$$c(2,(2,1,2,1)=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ a_{11} & a_{12} & a_9 & a_{01} & a_{14} & a_{15} & a_{13} & a_3 & a_4 & a_1 & a_2 & a_6 & a_7 & a_8 & a_5 \end{pmatrix}$$

Теперь представим, что алгоритм перестановки, определенный для множества, соответствующего одному атрибуту, проводится для всех множеств атрибутов исходной таблицы. В этом случае полный алгоритм перестановки задается следующим набором параметров:

1. $(K_1, K_2, ..., K_N)$ - множество, определяющее количество подмножеств для множества каждого атрибута, которое определяет подмножества элементов $(U_{11}, U_{12}, ..., U_{1K_i}), (U_{21}, U_{22}, ..., U_{2K_2}), ..., (U_{N1}, U_{N2}, ..., U_{NK_N})$;

2. $((M_{11}, M_{12}, ..., M_{1K_i}), (M_{21}, M_{22}, ..., M_{2K_2}), ..., (M_{N1}, M_{N2}, ..., M_{NK_N}))$ - множество, определяющее число элементов в подмножествах для множества каждого атрибута;

3. $((r_{01}, \mathbf{r}_1), (r_{02}, \mathbf{r}_2), ..., (r_{0N}, \mathbf{r}_N))$ - множество параметров перестановок для множества каждого атрибута.

Этот набор задает параметры процедуры перемешивания для исходной таблицы $T(t_1, t_2, ..., t_N)$.

В результате применения процедуры, вместо исходной таблицы $T(t_1, t_2, ..., t_N)$ получается таблица обезличенных данных $\widetilde{T}(t_1, t_2, ..., t_N)$.

Набор параметров:

$$G(T(t_1, t_2, ..., t_N)) = \{(K_1, K_2, ..., K_N),$$
$$((M_{11}, M_{12}, ..., M_{1K_i}), (M_{21}, M_{22}, ..., M_{2K_2}), ..., (M_{N1}, M_{N2}, ..., M_{NK_{N)}}),$$
$$((r_{01}, \mathbf{r}_1), (r_{02}, \mathbf{r}_2), ..., (r_{0N}, \mathbf{r}_N))\}$$

полностью и однозначно задает процедуру перемешивания для исходной таблицы $T(t_1, t_2, ..., t_N)$.

Пример 2. Пусть исходная таблица $T(t_1,t_2,t_3,t_4)$ имеет вид:

Атрибут t_1	**Атрибут** t_2	**Атрибут** t_3	**Атрибут** t_4
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2
a_3	b_3	c_3	d_3
a_4	b_4	c_4	d_4
a_5	b_5	c_5	d_5
a_6	b_6	c_6	d_6
a_7	b_7	c_7	d_7
a_8	b_8	c_8	d_8
a_9	b_9	c_9	d_9
a_{10}	b_{10}	c_{10}	d_{10}
a_{11}	b_{11}	c_{11}	d_{11}
a_{12}	b_{12}	c_{12}	d_{12}
a_{13}	b_{13}	c_{13}	d_{13}
a_{14}	b_{14}	c_{14}	d_{14}
a_{15}	b_{15}	c_{15}	d_{15}

Для этой таблицы заданы следующие параметры процедуры перемешивания:

$G(T(t_1,t_2,t_3,t_4)) = \{(4,3,4,4),((4,4,4,3),(5,5,5),(6,3,3,3),(4,4,3,4)),$

$((2,(2,1,2,1)),(1,(3,4,2)),(2(2,1,2,1)),(3,(3,2,1,2))\}.$

После проведения процедуры перемешивания получаем таблицу $\widetilde{T}(t_1,t_2,t_3,t_4)$:

Атрибут t_1	**Атрибут** t_2	**Атрибут** t_3	**Атрибут** t_4
a_{11}	b_{14}	c_{11}	d_7
a_{12}	b_{15}	c_{12}	d_8
a_9	b_{11}	c_{10}	d_5
a_{10}	b_{12}	c_{15}	d_6
a_{14}	b_{13}	c_{13}	d_{11}
a_{15}	b_3	c_{14}	d_9
a_{13}	b_4	c_6	d_{10}
a_3	b_5	c_5	d_{14}
a_4	b_1	c_1	d_{15}
a_1	b_2	c_2	d_{12}
a_2	b_9	c_3	d_{13}
a_6	b_{10}	c_4	d_2
a_7	b_6	c_9	d_3
a_8	b_7	c_7	d_4
a_5	b_8	c_8	d_1

Как видно из примера, получена преобразованная таблица, в которой записи не соответствуют записям в исходной таблице, что обеспечивает достаточно высокую сложность восстановления исходной таблицы при отсутствии сведений о параметрах процедуры перемешивания.

Де-обезличивание.

Для практического применения указанной процедуры обезличивания необходимо иметь возможность формировать правильные записи (соответствующие записям в исходной таблице).

Пусть в столбце атрибута t_i таблицы $\widetilde{T}(t_1,t_2,...,t_N)$ выбран элемент номер n_i, тогда из матрицы результирующей перестановки $c_{0i}(r_{oi})$ можно получить номер этого элемента в исходной таблице - $m_i(n_i)$, который находится как элемент второй строки столбца номер n_i. Далее, в каждом столбце атрибута *j*, в соответствии с матрицей результирующей перестановки $c_{0j}(r_{oj})$, находится элемент, номер которого равен номеру столбца, во второй строке которого стоит число $m_i(n_i)$ (номер элемента в исходной таблице). Таким образом, после просмотра всех столбцов таблицы $\widetilde{T}(t_1,t_2,...,t_N)$ будет построена запись, соответствующая элементу номер n_i из множества атрибута t_i, соответствующая записи номер $m_i(n_i)$ в таблице $T(t_1,t_2,...,t_N)$.

Для оценки защищенности предложенной процедуры обезличивания используем такую характеристику, как число вариантов обезличивания, получаемых при применении данной процедуры.

Число возможных различных вариантов разбиения множества из *M* элементов на K_i подмножеств, удовлетворяющих условиям разбиения, приведенным выше, при заданном наборе $(M_{i1},M_{i2},...,M_{iK_i})$ равно $(K_i)!$ (при условии, что все подмножества имеют различное число элементов).

Максимальное число возможных вариантов для заданного набора разбиений *N* множеств атрибутов:

$$R((K_1,K_2,...,K_N),(r_{01},\mathbf{r}_1),(r_{02},\mathbf{r}_2),...,(r_{0N},\mathbf{r}_N))=\prod_{i=1}^{N}(K_i)!(K_i-1)(M_{i1}-1)(M_{i2}-1)...(M_{iK_i}-1).$$

При большом числе записей число вариантов получается очень большим, что обеспечивает очень малую вероятность подбора параметров, т.е. хорошую защиту обезличенных данных.

Заключение.

Перемешивание данных предложенным методом реализуется достаточно простыми средствами и может применяться в уже сформированных базах данных.

Большое количество вариантов параметров перемешивания обеспечивает достаточно эффективную защиту от атаки путем подбора параметров.

Использование процедуры перемешивания для обезличивания данных обеспечивает защиту от атак, использующих внешние данные, имеющие наборы атрибутов, совпадающие с некоторыми атрибутами в исходной таблице [5].

Наличие набора записей из исходной таблицы не позволяет провести процедуру де-обезличивания для других записей из таблицы обезличенных данных.

Однако, применение процедур перемешивания, предлагаемых в настоящей статье, в конкретной системе, должно учитывать правовые последствия использования подмножества (конечного множества значений) персональных данных, а каждый критерий разбиения должен быть согласован с технологией обработки персональных данных в указанной системе и ее окружении. Оценку эффективности обезличивания необходимо проводить для каждой реализации конкретной системы, включая рабочие места с ограниченным доступом и вывод информации на внешние носители.

Литература

1. Федеральнй закон «О персональных даннх»,- 2-е изд. – М.: «Ось-89», 2008. – 32 с.
2. McCallister E., Grance T., Scarfone K. Guide to Protecting the Confidentiality of Personally Identifiable Information (PII). Recomendations of the National Institute of Standarda and Technology (NIST) U.S. 2010.
3. Конопкин Н. Как превратить предприятие в легитимного менеджера персональных данных // IT – Manager, 11, 2009.
4. Калужин Л.А., Сущанский В.И. Преобразования и перестановки. – М.: Наука, 1985. – 160 с.
5. L. Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002. - P. 557-570.
6. http://ispdn.ru/forum/index.php?PAGE_NAME=read&FID=1&TID=1161 (дата обращения 10.01.2011).
7. http://xp-7.ru/blog/2010-01-03-26 (дата обращения 27.01.2011).