УДК 004.056

АЛГОРИТМ ДЕПЕРСОНАЛИЗАЦИИ ПЕРСОНАЛЬНЫХ ДАННЫХ

А.С. Куракин

Рассматривается проблема обеспечения безопасности информационных систем персональных данных на стадиях разработки и оптимизации. Для решения данной проблемы предлагается использовать алгоритм деперсонализации для понижения класса информационных систем персональных данных, что, в свою очередь, снимает основную часть требований по применению организационных мер и технических средств защиты информации.

Ключевые слова: обезличивание, алгоритм деперсонализации, защита информации, персональные данные, информационные системы персональных данных

Введение

Требования Федерального закона «О персональных данных» [1] и постановления Правительства Российской Федерации «Об утверждении Положения об обеспечении безопасности персональных данных при их обработке в информационных системах персональных данных» [2] обязывает организации, обрабатывающие персональные данные (далее – операторы), обеспечивать соответствующую информационную безопасность персональных данных (ПДн). В свою очередь, выполнение данных требований влечет за собой значительные материальные затраты на внедрение дополнительных средств защиты информации, что зачастую не предусмотрено бюджетом операторов. Альтернативным законным способом решения данной проблемы является обезличивание персональных данных, так как оно позволяет снизить требования к уровню защищенности данных, что влечет за собой соответствующее сокращение расходов на обеспечение их информационной безопасности. Под обезличиванием персональных данных, как правило, понимают алгоритмы, в результате выполнения которых невозможно определить принадлежность персональных данных их владельцу.

Наиболее часто на практике применяются следующие способы обезличивания персональных данных [3]:

1. уменьшение перечня сведений, подлежащих автоматизированной обработке; в результате получают перечень данных, который соответствует оптимальному объему персональных данных о каждом субъекте, необходимому для хранения в информационных системах персональных данных (ИСПДн);
2. замена идентификаторами части сведений, что позволяет понизить класс ИСПДн, так как система оперирует не с персональными данными субъектов напрямую, а с идентификационной информацией, лишенной содержательного контекста;

3. уменьшение детализации некоторых сведений; способ направлен на то, чтобы сделать персональные данные субъектов менее точными. Это может достигаться, в том числе, путем группирования общих или непрерывных характеристик;
4. замена чисел минимальным, средним или максимальным значением; так как не всегда есть необходимость обрабатывать часть персональных данных каждого субъекта, то можно заменять данные минимальным, средним или максимальным значением по всей выборке или отдельным ее частям;
5. обработка групп сведений в разных информационных системах, для чего ИСПДн разделяются на взаимодействующие участки системы. Данный способ можно применять для оптимизации набора средств защиты информации (СЗИ), используемых в каждом сегменте. Это, с одной стороны, приводит к снижению стоимости защиты, а с другой – снижает избыточность СЗИ в тех случаях, когда защищаемые данные расположены неравномерно по системе.

Для всех перечисленных способов критерием качества деперсонализации является вероятность получения персональных данных на основании утечки обезличенных данных конкретного человека. При этом предполагается, что злоумышленнику известен контекст обработки, а также дополнительная информация из других источников.

Основным недостатком указанных способов является то, что они не гарантируют отсутствие возможности получения персональной информации посредством контекстного анализа открытой информации, в том числе получаемой из смежных систем.

Целью данной работы является разработка стойкого алгоритма обезличивания персональных данных (далее по тексту – алгоритма деперсонализации), основанного на применении перестановок и позволяющего проводить обезличивание больших массивов персональных данных при минимальных объемах служебной информации.

Алгоритм деперсонализации

Перспективным способом решения поставленной задачи является перестановка персональных данных, относящихся к различным субъектам. Данный способ обладает следующими преимуществами: персональные данные хранятся в одной ИСПДн, значительно снижается вероятность успеха контекстного анализа.

Предлагаемый алгоритм деперсонализации построен на следующих принципах:

1. разбиение исходного множества данных на подмножества [4], что позволяет сократить размерность и упростить его практическую реализацию;
2. использование циклических перестановок [4], что реализует собственно перемешивание данных.

В качестве исходных данных возьмем таблицу персональных данных $D(d_1, d_2, \ldots, d_N)$, где N – число атрибутов, а M – число строк таблицы.

Далее рассмотрим множество данных, относящееся к одному атрибуту – d_i (i=1, 2, …, N). Это множество атрибута d_i – A_i, содержит M элементов. Все элементы каждого множества A_i пронумерованы от 1 до M, и в таблице $D(d_1, d_2, \ldots, d_N)$ совокупность элементов множеств разных атрибутов с одинаковыми номерами будем называть записью с соответствующим номером. При этом 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.

Алгоритм обеспечивает перестановку данных каждого множества атрибутов исходной таблицы пошагово. На каждом шаге используется принцип циклических перестановок.

Проведем разбиение множества A_i на A_i ($M>A_i>1$) непересекающихся подмножеств A_{ij}, где число элементов подмножества A_{ij} равно M_{ij} ($M> M_{ij\,i}>1$), j=1, 2, …, K_i. Все элементы каждого подножества A_{ij} считаем занумерованными от 1 до M_{ij}, эти номера будем называть внутренними номерами элементов подмножества. Внешний номер элемента в подмножестве A_{ij}, имеющего внутренний номер k, обозначим – m_{ijk} ($1\le m_{ijk} \le M$), где m_{ijk} – это порядковый номер элемента на множестве A_i, соответствующий элементу с внутренним номером k. Разбиение каждого множества должно обладать следующими свойствами:

1. $A_i = \bigcup_{j=1}^{K_i} A_{ij}$ – подмножества разбиения включают все элементы множества A_i;
2. $A_{ij} \neq \varnothing$, $A_{ij} \cap A_{im} = \varnothing$ для всех j, m =1, 2, …, K_i – каждое подмножество не пусто, а пересечение любых двух подмножеств пусто;
3. $m_{ij1}= m_{i(j\text{-}1)Mi(j\text{-}1)}+1$ для всех j = 2, …, K_i – для любых двух подмножеств A_{ij} и $A_{i(j\text{-}1)}$ элемент с первым внутренним номером подмножества A_{ij} имеет на единицу больший внешний номер, чем внешний номер элемента с наибольшим внутренним номером подмножества $A_{i(j\text{-}1)}$;
4. если $k_1>k_2$, то $m_{ijk1}> m_{ijk2}$ для всех i =1, 2, …, N; j =1, 2, …, K_i – упорядоченность внешней и внутренней нумераций для всех множеств и подмножеств их разбиения совпадают;

5. $M = \sum_{j=1}^{K_i} M_{ij}$ – суммарное число элементов всех подмножеств A_{ij} равно общему числу элементов множества A_i.

Для каждого подмножества A_{ij} определим циклическую перестановку (подстановку) p_{ij} (r_{ij}), задаваемую следующим образом:

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

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

Таким образом, в перестановке (подстановке) p_{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 $p_{ij}(r_{ij})$. Данный параметр задается генератором случайных чисел (ГСЧ) в интервале [1; $M_{ij} - 1$]. Теперь все перестановки для всех подмножеств множества A_i можно задать набором (вектором) параметров r_i=(r_{i1}, r_{i2}, …, r_{iKi}). Вектор параметров перестановок r_i задает первый уровень алгоритма деперсонализации, т.е. перестановки первого уровня.

Рассмотрим теперь множество a_i=(a_{i1}, a_{i2}, …, a_{iKi}), состоящее из K_i элементов. Здесь элемент a_{ij} соответствует подмножеству A_{ij}, j =2, 3, …, K_i. Для этого множества определим циклическую перестановку $p_{0j}(r_{0j})$:

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

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

Таким образом, в перестановке p_{0j} (r_{0j}) производится циклический сдвиг элементов множества a_i (подмножеств множества A_i) на число r_{0i} ($1 \le r_{0i} \le K_i - 1$) – параметр перестановки. Данный параметр r_{0i} задается ГСЧ в интервале [1; $K_i - 1$]. Эту перестановку будем называть перестановкой второго уровня.

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

Определим теперь нумерацию элементов множества A_i после проведения всех перестановок. С учетом правил перемножения перестановок имеем следующую результирующую перестановку:

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

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

Примеры реализации алгоритма

Пример 1. Пусть M=15, K_i=4 и M_{i1}= 4, M_{i2}= 4, M_{i3}= 4, M_{i4}= 3, при этом
d_i=(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}),
A_{i1}=(b_1, b_2, b_3, b_4), A_{i2}=(b_5, b_6, b_7, b_8), A_{i3}=(b_9, b_{10}, b_{11}, b_{12}), A_{i4}=(b_{13}, b_{14}, b_{15}),
r_i=(2, 1, 2, 1), r0$_i$=2.
После применения перестановок первого уровня имеем

$$p_{i1}(2)=\begin{pmatrix}1&2&3&4\\ b_3&b_4&b_1&b_2\end{pmatrix},\ p_{i1}(1)=\begin{pmatrix}1&2&3&4\\ b_6&b_7&b_8&b_5\end{pmatrix},$$

$$p_{i2}(2)=\begin{pmatrix}1&2&3&4\\ b_{11}&b_{12}&b_9&b_{10}\end{pmatrix},\ p_{i4}(1)=\begin{pmatrix}1&2&3\\ b_{14}&b_{15}&b_{13}\end{pmatrix}.$$

Результирующая перестановка имеет вид

$$p(2,(2,1,2,1))=\begin{pmatrix}1&2&3&4&5&6&7&8&9&10&11&12&13&14&15\\ b_{11}&b_{12}&b_9&b_{10}&b_{14}&b_{15}&b_{13}&b_8&b_4&b_1&b_2&b_6&b_7&b_9&b_5\end{pmatrix}.$$

Теперь представим, что алгоритм перестановки, определенный для множества, соответствующего одному атрибуту, применяется ко всем множествам атрибутов исходной таблицы. В этом случае полный алгоритм перестановки задается следующим набором параметров:

1. $(K_1, K_2, \ldots, K_N)$ – множество, определяющее количество подмножеств для множества каждого атрибута, которое определяет подмножества элементов $(A_{11}, A_{12}, \ldots, A_{1K1})$, $(A_{21}, A_{22}, \ldots, A_{2K2})$, …, $(A_{N1}, A_{N2}, \ldots, A_{NKN})$.
2. $(M_{11}, M_{12}, \ldots, M_{1K1})$, $(M_{21}, M_{22}, \ldots, M_{2K2})$, …, $(M_{N1}, M_{N2}, \ldots, M_{NKN})$ – множество, определяющее число элементов в подмножествах для множества каждого атрибута;
3. $((r_{01}, r_1), (r_{02}, r_2), \ldots, (r_{0N}, r_N)$ – множество параметров перестановок для множества каждого атрибута.

Этот набор задает параметры алгоритма деперсонализации для исходной таблицы $D(d_1, d_2, \ldots, d_N)$.

В результате применения процедуры вместо исходной таблицы $D(d_1, d_2, \ldots, d_N)$ получается таблица обезличенных данных $\breve{D}(d_1, d_2, ..., d_N)$. Набор параметров

$$C\left(D\left(d_1,d_2,...,d_N\right)\right)=\left\{\left(K_1,K_2,...,K_N\right),\left(\left(M_{11},M_{12},...,M_{1K_1}\right),\left(M_{21},M_{22},...,M_{2K_2}\right),...,\left(M_{N1},M_{N2},...,M_{NK_N}\right)\right),\right.$$
$$\left.\left(\left(r_{01},r_1\right),\left(r_{02},r_2\right),...,\left(r_{0N},r_N\right)\right)\right\}.$$

полностью и однозначно задает алгоритм деперсонализации для исходной таблицы $D(d_1, d_2, \ldots, d_N)$.

Пример 2. Пусть исходная таблица $D(d_1, d_2, \ldots, d_N)$ имеет вид, представленный в табл. 1.

Атрибут d_1	Атрибут d_2	Атрибут d_3	Атрибут d_4	Атрибут d_5	Атрибут d_6
q_1	r_1	s_1	t_1	u_1	v_1
q_2	r_2	s_2	t_2	u_2	v_2
q_3	r_3	s_3	t_3	u_3	v_3
q_4	r_4	s_4	t_4	u_4	v_4
q_5	r_5	s_5	t_5	u_5	v_5
q_6	r_6	s_6	t_6	u_6	v_6
q_7	r_7	s_7	t_7	u_7	v_7
q_8	r_8	s_8	t_8	u_8	v_8
q_9	r_9	s_9	t_9	u_9	v_9
q_{10}	r_{10}	s_{10}	t_{10}	u_{10}	v_{10}

Таблица 1. Исходная таблица данных

Для этой таблицы заданы следующие параметры алгоритма деперсонализации:

$$C\left(D\left(d_1,d_2,d_3,d_4,d_5,d_6\right)\right)=\left\{\left(3,2,4,3,3,2\right),\left(\left(3,3,4\right),\left(6,4\right),\left(2,3,2,3\right),\left(3,4,3\right),\left(5,2,3\right),\left(3,7\right)\right),\right.$$
$$\left.\left(\left(2,(1,2,3)\right),\left(1,(3,1)\right),\left(3,(1,2,1,1)\right),\left(2,(2,1,2)\right),\left(2,(4,1,1)\right),\left(1,(1,4)\right)\right)\right\}.$$

После выполнения алгоритма деперсонализации получаем таблицу $\breve{D}(d_1, d_2, ..., d_N)$ (табл. 2).

Как видно из примера, в результате применения алгоритма деперсонализации 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 алгоритма деперсонализации. Доступность персональных данных (получение достоверных персональных сведений при легитимном обращении к ним) обеспечивается посредством решения обратного алгоритма деперсонализации, результатом чего является формирование исходной таблицы.

Атрибут d_1	Атрибут d_2	Атрибут d_3	Атрибут d_4	Атрибут d_5	Атрибут d_6
q_{10}	r_8	s_9	t_{10}	u_9	v_8
q_7	r_9	s_{10}	t_8	u_{10}	v_9
q_8	r_{10}	s_8	t_9	u_8	v_{10}
q_9	r_7	s_2	t_3	u_5	v_4

q_2	r_4	s_1	t_1	u_1	v_5
q_3	r_5	s_5	t_2	u_2	v_6
q_1	r_6	s_3	t_5	u_3	v_7
q_6	r_1	s_4	t_6	u_4	v_2
q_4	r_2	s_7	t_7	u_7	v_3
q_5	r_3	s_6	t_4	u_6	v_1

Таблица 2. Результат второго шага

Реализация обратного алгоритма деперсонализации

Пусть в столбце атрибута d_i таблицы $\breve{D}(d_1, d_2, ..., d_N)$ (табл. 2) выбран элемент номер n_i, тогда из матрицы результирующей перестановки p_{0j} (r_{0j}) можно получить номер этого элемента в исходной таблице – $d_i(n_i)$, который находится как элемент второй строки столбца номер n_i. Далее в каждом столбце атрибута d_j в соответствии с матрицей результирующей перестановки p_{0j} (r_{0j}) 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 $\breve{D}(d_1, d_2, ..., d_N)$ будет построена запись, соответствующая элементу номер n_i из множества атрибута d_i и записи номер $m_i(n_i)$ в таблице $D(d_1, d_2, \ldots, d_N)$. (табл. 1).

Практическая реализация алгоритма

При практической реализации алгоритма деперсонализации контроль целостности данных в файле обеспечивается путем проверки текущей контрольной суммы всего файла (сформированной при сохранении (модификации) файла) и контрольной суммы, рассчитываемой при последующем открытии файла. Как правило, данные алгоритмы контроля целостности хранимой в постоянном запоминающим устройстве (ПЗУ) информации (файлов) реализованы в механизмах защиты операционной системы, аппаратно-программном модуле доверенной загрузки (АПМДЗ) или специализированном программном обеспечении – СЗИ от несанкционированного доступа (НСД).

Перспективным развитием алгоритма деперсонализации в части обеспечения контроля целостности информации (персональных данных) является интеграция механизма формирования имитовставки, что обеспечит, помимо контроля целостности, более высокую степень защиты от НСД.

В качестве программной реализации алгоритма деперсонализации ПДн на языке C# предлагается экспериментальный образец программного обеспечения (ЭО ПО) «Depersonalization». Данное ПО работает в двух режимах:

– режим 1: программа производит деперсонализацию исходных данных;
– режим 2: программа осуществляет обратный алгоритм деперсонализации, приводя данные к исходному виду.

Оценка защищенности алгоритма деперсонализации

Для оценки защищенности предложенного алгоритма деперсонализации используем такую характеристику, как число вариантов деперсонализации, получаемых при применении данного алгоритма.

Число возможных различных вариантов разбиения множества из M элементов на K_i подмножеств, удовлетворяющих условиям разбиения, приведенным выше, при заданном наборе $\left(M_{i1}, M_{i2}, ..., M_{iK_N}\right)$ равно $(K_i)!$ (при условии, что все подмножества имеют различное число элементов). Максимальное число возможных вариантов для заданного набора разбиений N множеств атрибутов равно

$$V\left(\left(K_1, K_2, ..., K_N\right), \left(M_{11}, M_{12}, ..., M_{1K_1}\right), ..., \left(M_{i1}, M_{i2}, ..., M_{iK_N}\right)\right) =$$

$$= \prod_{i=1}^{N} \left(K_i\right)! \left(K_i - 1\right)\left(M_{i1} - 1\right)\left(M_{i2} - 1\right)...\left(M_{iK_i} - 1\right).$$

При большом количестве записей число вариантов получается очень большим, что обеспечивает очень малую вероятность подбора параметров и соответственно хорошую защиту обезличенных данных.

Для числовой оценки рассчитаем число возможных вариантов разбиения на примере ИСПДн, в которой одновременно обрабатываются паспортные данные 100 субъектов в пределах конкретной организации:

- фамилия;

- имя;
- отчество;
- серия и номер паспорта;
- дата рождения;
- пол;
- адрес.

В данном примере для простоты вычислений зададим одинаковые параметры разбиений для всех множеств атрибутов: $K_i = 10, i = \overline{1, N}, N = 7$. Мощности подмножеств распределим следующим образом: M_1=5, M_2=6, M_3=7, M_4=8, M_5=9, M_6=11, M_7=12, M_8=13, M_9=14, M_{10}=15.

В результате вычислений по вышеуказанной формуле получаем максимальное число возможных вариантов для заданного набора разбиений, равное $1{,}13\times10^{117}$. Таким образом, в случае, когда злоумышленник, имея деперсонализированные персональные сведения, применяет для получения достоверной информации метод «грубой силы» (полного перебора) [5], ему требуется более $1{,}19\times10^{108}$ лет при вычислительных ресурсах, обеспечивающих скорость перебора 3000000 вариантов в секунду.

Заключение

Предложенный алгоритм деперсонализации является перспективным и оптимальным решением задач по обеспечению информационной безопасности персональных данных, обрабатываемых в информационных системах персональных данных.

Данный алгоритм обладает следующими преимуществами:

- обеспечивает защиту персональных сведений от несанкционированного доступа, в том числе от компрометации информации при ее утечке по техническим каналам;
- обеспечивает гарантированный доступ к персональным данным при легитимном обращении;
- персональные сведения хранятся в одной таблице;
- получение персональных сведений посредством контекстного анализа или путем перебора весьма трудоемко, а зачастую практически невозможно;
- параметры перестановки задаются при помощи генератора случайных чисел, что увеличивает стойкость алгоритма к взлому.

Наибольшая эффективность при применении данного алгоритма проявляется в случае, когда в информационных системах персональных данных содержатся большое количество персональных данных субъектов, что обеспечивает наибольшую защиту информационной системе.

Литература

1. Федеральный закон Российской Федерации от 27 июля 2006 г. № 152-ФЗ. О персональных данных. Принят Государственной Думой Федерального Собрания Российской Федерации 8 июля 2006 г.: одобрен Советом Федерации Федерального Собрания Российской Федерации 14 июля 2006 г. // Российская газета. – 2006. – 29 июля.
2. Об утверждении Положения об обеспечении безопасности персональных данных при их обработке в информационных системах персональных данных: Постановление Правительства Российской Федерации от 17 ноября 2007 г. № 781 г. Москва // Российская газета. – 2007. – 21 ноября.
3. Царев Е. Информационная безопасность по-русски // День 4. Обезличивание персональных данных. 2009 – [Электронный ресурс]. – URL: http://www.tsarev.biz/news/den-4-obezlichivanie-personalnyx-dannyx, свободный. Яз. рус. (дата обращения: 29.03.12).
4. Стенли Р. Перечислительная комбинаторика. – М.: Мир, 1990. – 440 с.
5. Полный перебор // Википедия – свободная энциклопедия – [Электронный ресурс]. – URL: http://ru.wikipedia.org/wiki, свободный. Яз. рус. (дата обращения: 15.04.2012).

Куракин Александр Сергеевич – Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, аспирант, nirtit@gmail.com