А. В. Балакин, **А. С. Елисеев** (Ростов-на-Дону, ФГНУ НИИ «Спецвузавтоматика»). **Метод защиты базы данных от несанкционированного копирования.**

Несмотря на все системы защиты, в современных СУБД из-за небрежности и ненадежности персонала конфиденциальная информация часто становится общедоступной. Единственным способом защиты от этого является принципиальная невозможность восстановления и просмотра конфиденциальных данных вне их исходного хранилища.

Для этого предлагается сделать физическое местоположение данных БД на носителе информации неотъемлемой частью этих данных. Суть такой привязки может быть основана на том, что при переносе данных между хранилищами легко перенести сами данные, однако как минимум весьма сложно, и часто невозможно, оставить физическое положение этих данных на носителе неизменным.

Способы сделать физический адрес данных частью самих данных могут быть различными. В общем случае у нас должно быть некоторое отображение f, такое что:

$$x = f(s, k),$$

где s — данные, хранимые на носителе, k — адрес этих данных на нем, x — полезные данные, которые мы защищаем. В БД для получения данных x, необходимо вызывать $x = f(s, k)$. Для того чтобы сделать эту операцию прозрачной можно использовать представление, а если f вычисляется долго, а данные изменяются редко, то целесообразно создать материализованное представление. Также, необходимо некоторое отображение g, такое что:

$$s = g(x, k).$$

Оно должно применяться для внесения изменений в БД и для добавления новых записей.

Заметим, что поскольку данные одной таблицы обычно хранятся рядом, то адреса соседних строк БД на носителе отличаются лишь незначительно. Поэтому для надежной защиты отображение $s = g(x, k)$ должно обладать неустойчивостью по k. Это же требование предъявляется к криптосистемам, поэтому в качестве отображений g и f можно использовать соответственно функции зашифрования и расшифрования какой-либо блочной криптосистемы, в качестве ключа которых необходимо использовать адрес k.

Данный метод был успешно реализован для СУБД Oracle. Реализация основана на том, что при написании запросов в этой СУБД, можно для каждой таблицы использовать псевдостолбец ROWID, который возвращает идентификатор, часть которого основана на адресе данной строки на физическом носителе. Кроме того, Oracle гарантирует, что идентификатор строки стандартной (не индексной) таблицы, организованной по принципу кучи, не меняется в процессе работы СУБД.

К недостаткам предложенного метода можно отнести накладываемое им требование неизменности местоположения данных на носителе в процессе работы СУБД, некоторое замедление извлечения и изменения данных, а также усложнение санкционированного переноса данных между хранилищами. Однако, при этом, он позволяет предотвратить обнародование конфиденциальной информации хранящейся в БД даже в условиях неблагонадежности персонала, работающего с этой БД.

СПИСОК ЛИТЕРАТУРЫ

1. *101 Oracle.* Резервное копирование и восстановление, 2005, 464 с.
2. *Кайт Т.* Oracle для профессионалов. СПб.: ООО «ДиаСофтЮП», 2003, 672 с.