Изобретение относится к карточкам с микросхемами, а более конкретно - к карточкам с микропроцессорами, которые сами вносят различные изменения в свою собственную энергонезависимую память.

При выполнении транзакции (деловой операции), память в общем случае изменяется один или несколько раз и, естественно, необходимо гарантировать, что все изменения были выполнены правильно, до наступления того момента, когда понадобится использовать вновь записанную информацию, и что эта вновь записанная информация будет проигнорирована или стерта в случае ошибки или записи, страдающей искажениями.

В патенте США № 4877945A описано, как обнаружить аномалию, которая возникает при осуществлении последовательности записи множества элементов данных, чтобы предотвратить продолжение транзакции на ошибочной основе.

В случае аномалии, также целесообразно иметь возможность восстановить статус-кво, т.е. обеспечить для следующей транзакции такое положение, чтобы можно было обрабатывать значения информации, которые были записаны в карточку, до осуществления неправильной транзакции.

В вышеупомянутом патенте США № 4877945A это преимущество не предлагается, поскольку в некоторых случаях старые значения информации будут потеряны во время осуществления неправильной транзакции, так что будет невозможно восстановить упомянутую информацию до ее ранее существовавшего состояния; по меньшей мере, это будет не просто сделать на основе информации, содержащейся в карточке.

В международной заявке WO-A-89/02140 описан один такой способ работы, но он применим только в случае, когда изменен один элемент информации или когда изменяют множество элементов информации независимо друг от друга.

Однако во многих случаях необходимо изменять множество элементов информации в течение одной транзакции, и их считают "взаимозависимыми", когда их нужно обрабатывать вместе, чтобы гарантировать, что все изменения внесены во множество элементов информации надлежащим образом.

Риск несовершенной или незавершенной транзакции, связанный с множеством независимых элементов информации, является высоким, в частности, в случае карточек "бесконтактного" типа, в которых не ощущаются границы объема, в пределах которого карточка может правильно работать в терминале. При таких обстоятельствах существует значительный риск неожиданного прерывания связи между карточкой и терминалом до окончания обработки или вследствие переходного возмущения, например - прохождения некоторой массы металла рядом.

Примером (который, естественно, не является ограничительным) служит применение такой карточки в транзакции дистанционного заказа билетов, т.е. для доступа в общественную транспортную сеть, когда карточка играет две роли: роль, присущую билету для проезда, и роль, присущую электронному кошельку.

Уже предложены несколько решений, преодолевающих в какой-то степени вышеупомянутые трудности и способствующих "неделимому" внесению множества записей или других изменений в независимые элементы данных.

Например, в конкретном приложении, о котором шла речь выше, известные системы начинают работу дебетованием кошелька, а затем они регистрируют права на проезд, приобретенные пользователем. Если пользователь извлекает карточку между этими двумя операциями, то пользователя приглашают представить карточку снова и повторно начинают запись прав на проезд. Однако, если пользователь уйдет, не представляя карточку снова, то он останется ни с чем. Очевидно, что невозможно действовать в обратном порядке, поскольку пользователь тогда попытался бы забрать карточку до дебетования кошелька.

Такое решение подразумевает, что терминал специально сконфигурирован с возможностью - в случае прерывания - активизировать проведение обработки для управления повторным началом транзакции (повторной вставкой карточки в терминал по запросу). Помимо усложнения программного обеспечения терминала, такое решение не вполне удовлетворительно, поскольку, как упоминалось, пользователь попрежнему может остаться ни с чем в случае, если транзация не начинается снова.

Другое решение заключается в пересечении (просмотре) данных, когда терминал сохраняет информацию о состоянии кошелька в карточке, и наоборот. Однако это решение неудовлетворительно в любом случае, потому что, помимо своей сложности, оно увеличивает объем данных, которыми обмениваются карточка и терминал, а значит - замедляет проведение транзакции. Его также трудно применить, когда большое число (три или более) записей приходится делать неделимыми.

Одна из задач изобретения состоит в том, чтобы предложить способ, обеспечивающий внесение множества изменений в памяти карточки неделимым образом.

Другая задача изобретения состоит в том, чтобы предложить такой способ, которым могла бы полностью управлять карточка. Таким образом, этот способ может быть реализован без изменения терминалов и без какой-либо потребности в проведении обработки в терминалах, при этом способ использует синтаксис существующих команд и поэтому является очень

гибким в смысле команд, которые можно выбирать.

Способ, согласно изобретению является способом того типа, при котором карточку временно подключают к терминалу во время проведения транзакции, причем транзакция включает в себя подачу из терминала в карточку множества команд изменения, каждая из которых содержит, по меньшей мере, одну операцию записи в память карточки соответствующего элемента данных, обозначенного этой командой, при этом различные элементы данных, записываемые таким образом, являются взаимозависимыми. Способ согласно изобретению отличается тем, что включает выполнение карточкой следующих этапов: а) после соответствующего приема соответствующих команд из терминала она изменяет содержимое памяти карточки путем условной записи в памяти карточки каждого из упомянутых независимых элементов данных информации без потери предыдущих значений, соответствующих упомянутым элементам, а затем б) оканчивает изменения, либо все их подтверждая, либо все их отклоняя, так что при последующих операциях команды, выполненные на этапе а), либо будут все учтены, либо все будут безрезультатными.

Таким образом, принцип, на котором основано изобретение, заключается в том, что группируют воедино множества изменений, которые нужно внести неделимым образом за один этап а), а затем - после осуществления изменений в карточке верифицируют все эти изменения. Если верификация успешна, то на следующей операции, проводимой карточкой (во время той же транзакции или во время последующей транзакции), доступное содержимое обязательно будет отражать изменения, которые внесены.

И наоборот, любое прерывание в работе карточки, имеющее место на этапе а), уничтожит все внесенные изменения, и данные в энергонезависимой памяти останутся в состоянии, имевшем место перед этапом а).

В конкретной реализации способа согласно изобретению в случае подтверждения на этапе б) флаг, подтверждающий надлежащее выполнение, записывают в памяти карточки, а когда карточка впоследствии принимает команду, требующую осуществить считывание или изменение, по меньшей мере, одного из элементов данных, записанных на этапе а), или соответствующего ему значения, карточка начинает работать, проверяя состояние флага, и если он не записан, карточка игнорирует или удаляет условные записи, сделанные ранее на этапе а), и выполняет команду на основе упомянутых предыдущих значений, соответствующих элементам данных. Если во время проверки карточкой состояния флага обнаруживается, что он записан, то карточка может выполнять операции копирования условных записей, сделанных на этапе а).

В наиболее предпочтительном варианте реализации изобретения карточка пригодна для работы в двух режимах, а именно: во внутрисеансовом режиме, в котором записи делают путем выполнения этапов а) и б), и во внесеансовом режиме, в котором внесение записей не подтверждается для всех этапов а) и б).

Начало сеанса может быть неявным, например, путем обнуления карточки, или оно может последовать за командой, вызывающей два действия: выполнение предварительно определенной операции и интерпретацию в качестве команды начать сеанс.

Например, когда обычно сертифицируемая запись не сопровождается сертификатом, карточка автоматически начинает сеанс, во время которого происходит обработка записи.

Точно так же завершение сеанса может быть неявным и происходить после команды, которая вызывает два действия: выполнение предварительно определенной операции и интерпретацию в качестве команды завершить сеанс.

Например, операция дебетования кошелька завершает сеанс, исключая таким образом любую необходимость задержать передачу получаемого сертификата и давая возможность сделать неотличимыми сертификаты сеанса и сертификаты транзакций, осуществляемых с помощью кошельков.

В наиболее предпочтительном варианте реализации изобретения способ включает функцию аутентификации, объединенную с функцией завершения этапа б), за счет чего происходит принудительное отклонение этапа б) в случае безуспешной аутентификации.

В первом варианте реализации изобретения упомянутую аутентификацию выполняет карточка, которая аутентифицирует терминал и/или данные, которыми терминал обменивается с карточкой, причем карточка контролирует криптографический сертификат, создаваемый терминалом и передаваемый в карточку, и подтверждает изменения, внесенные на этапе б), только в случае, если сертификат признан правильным.

В режиме с сеансом можно предусмотреть такое условие, что, когда карточка принимает из терминала команды изменения содержимого памяти, включая создание криптографического сертификата, упомянутую верификацию проводят, если прием команды происходит вне сеанса, и не проводят, если прием команды происходит в сеансе.

Иными словами, те из команд, которые карточка выполняет на этапе б) и которые должны в обычных обстоятельствах (т.е. вне сеанса) верифицировать криптографический сертификат, больше не включают эту верификацию, когда их выполняют в сеансе, при этом

"сертификат сеанса, аутентифицирующий терминал" выполняет эквивалентную функцию.

Во втором варианте реализации изобретения упомянутую аутентификацию выполняет терминал, который аутентифицирует карточку и/или данные, которыми обмениваются терминал и карточка, причем карточка создает и передает криптографический сертификат обычным способом в терминал, если и только если изменения подтверждены на этапе б).

В режиме с сеансом можно предусмотреть такое условие, что, когда карточка принимает из терминала команды изменения содержимого памяти, включая создание криптографического сертификата, упомянутое создание осуществляют, если прием команды происходит вне сеанса, и не осуществляют, если прием команды происходит в сеансе.

Иными словами, те из команд, которые выполняются карточкой на этапе б) и которые должны в обычных обстоятельствах (т.е. вне сеанса) создавать криптографический сертификат, больше не включают это создание, когда они выполняются в сеансе, при этом "сертификат сеанса, аутентифицирующий терминал" выполняет эквивалентную функцию.

Предусмотрено и такое условие, что, когда карточка на этапе б) принимает из терминала команды изменения содержимого памяти, включая создание множества криптографических сертификатов, эти сертификаты запоминают на этапе б), а затем вместе передаются в терминал, только в случае, если изменения подтверждены на этапе б).

Иными словами, предусмотрено условие задержки передачи карточкой криптографических сертификатов, создаваемых по командам на этапе б). В частности, если сертифицированная команда записи создает некоторый сертификат записи, то будет целесообразным, чтобы сертификат покидал карточку только после осуществления записи без отмены.

В конкретном варианте реализации изобретения, по меньшей мере, некоторые из команд, которые могут быть выполнены на этапе б), включают в себя необязательный запрещающий атрибут, и если карточка выполняет такую команду в сеансе на этапе б), то изменения, вносимые по упомянутой команде, остаются в силе независимо от результата этапа б).

Иными словами, атрибут определяет, выполнялась ли команда в сеансе (т.е. будет аннулирована, если сеанс не завершен) или вне сеанса (т.е. все равно будет результативной, хотя и выполнена вне сеанса, даже если хронологически она проходит в сеансе).

В наиболее предпочтительном варианте реализации изобретения, после этапа б) и в случае подтверждаемых изменений, изобретение также предусматривает следующую последовательность этапов: г) терминал выполняет свою функцию после подтверждения карточкой, и д) в случае, если терминал выполнит упомянутую функцию надлежащим образом, информацию о ратификации записывают в карточку для последующего доступа путем считывания.

Такая "ратификация" сеанса информирует карточку о том, что терминал на самом деле оказался способным принимать решения (т.е. снимает препятствие в приложении, где речь идет о получении доступа в общественную транспортную сеть) с последующим проведением сеанса.

Следует отметить, что карточка управляет этой ратификацией, не нуждаясь в дополнительной записи (копировании условных записей, являющемся операцией, которую рано или поздно придется выполнить). Кроме того, это копирование выполняется на карточке лишь при условии, что на терминале надлежащим образом выполнено действие, т.е. только в случае, если вся транзакция удовлетворяет требованиям.

При выполнении всех операций, которыми управляет карточка, можно в предпочтительном варианте реализации изобретения предусмотреть условие выполнения команды записи на этапе д) как неявной команды, при этом любая команда, принимаемая карточкой после этапа б), интерпретируется как команда записи информации о ратификации в карточку.

Другие характеристики и преимущества станут ясны из нижеследующего описания двух вариантов реализаций изобретения.

В этих примерах и далее по всему тексту слово "обозначить" употребляется в смысле "задать один из множества" и относится к действию, которое заключается в характеристике некоторого конкретного элемента информации среди различных элементов информации, содержащихся в карточке.

Такое обозначение может быть неявным, поскольку сама команда задает конкретный элемент информации; например, команда "дебетовать сумму x из кошелька" обозначает ячейку памяти, которая содержит значение элемента данных "баланса кошелька".

Обозначение также может быть явным, как, например, в нижеследующем примере 1, где предусмотрено условие, согласно которому команды записи имеют адрес или идентификатор сектора, и при этом команды помечены индексом i.

Пример 1.

Предлагается карточка, которая запоминает 100 восьмибайтовых значений и может выполнить следующие команды:

- считывание восьмибайтового значения v как заданного индексом i в диапазоне 1-100;

- запись восьмибайтового значения v как заданного индексом i в диапазоне 1-100;

- начало сеанса;

- завершение сеанса.

Карточка должна обеспечивать проведение до трех записей в пределах одного сеанса.

Обычно для обозначения значений в энергонезависимой памяти (например, электрически стираемой программируемой постоянной памяти - ЭСППП) используют прописные буквы, а для обозначения значений в энергозависимой памяти (оперативной памяти или памяти с произвольным доступом - ППД, содержимое которой теряется при отключении питания) используют строчные буквы.

Одна зона энергонезависимой памяти выделена для запоминания основных данных карточки (окончательных записей):

- V[i], для i в диапазоне 1-100: 100 x 8 байт.

Другая зона энергонезависимой памяти выделена для запоминания механизма сеанса и содержит

- T[k], для j в диапазоне 1-3: 3x8 байт, содержащее значения, записанные во время сеанса (условные записи);

- I[k], для j в диапазоне 1-3: 3x8 байт, содержащее индексы, записанные во время сеанса; и

- C: счетный байт, который записывается в конце сеанса.

C кодирует число записей, сделанных в сеансе; подходящий механизм четности (например, связывающий дополнение упомянутого значения) дает возможность обнаружить случай, когда значение, запомненное в упомянутом счетном байте, является неопределенным.

Операции происходят следующим образом.

Этап 0: в некоторый момент между подачей электропитания в карточку и выполнением первой команды осуществляется проверка C. Если он представляет собой значение, которое определено в диапазоне 1-3, то, для k от 1 до C, из таблицы V[i] копируют значение T[k] с индексом I[k]. Затем C устанавливают равным нулю, а внутреннюю переменную j устанавливают равной -1 (чтобы указать, что сеанс не начался).

Этап 1: при считывании проводят тест, чтобы увидеть, соблюдается ли неравенство $j > 0$, и если оно соблюдается, то запрашиваемый индекс i сравнивают со значениями I[k] для k от j до 1 в убывающем порядке. Если имеет место совпадение, то возвращают T[k]. Во всех остальных случаях возвращают V[i].

Этап 2: в начале сеанса инициализируют j, устанавливая его на 0 (если сеанс уже начался, то этот этап аннулируется).

Этап 3: при каждой записи, если $j = -1$ (сеанс не начался), переданное значение v записывают в T[0], переданный индекс v записывают в I[0], а затем записывают $C = 1$, после чего v записывают в V[i] и записывают $C = 0$; если $0 \le j \le 3$ (запись в сеансе), то j увеличивают на 1, v записывают в T[j], а i записывают в I[j]; если $j = 3$, происходит отказ от операции (превышен предел записей в сеансе).

Этап 4: при закрытии сеанса, если $j > 0$, j записывают в C, а затем, для j от 1 до C, копируют значение T[j] с индексом I[j] из таблицы V[ ]. Затем C устанавливают равным 0, а j - равным -1.

Становится ясным, что хотя электропитание карточки может быть прервано в любой момент, и что считываемые значения будут правильными, т.е. для каждого индекса i это будет последнее значение, записанное не в сеансе, или записанное в сеансе, который завершен (запись завершена или сеанс завершен в момент, когда в C записали ненулевое значение).

Чтобы воспрепятствовать некоторым операциям, если криптографический сертификат, представляемый в карточку, является неправильным, и/или чтобы обеспечить выдачу криптографических сертификатов в карточку в конце некоторых операций, вводится криптография.

Используемые криптографические сертификаты основаны на известном типе криптографии. Например, "сертификат сеанса, аутентифицирующий карточку" (или терминал) получают путем применения защищенного алгоритма хеширования (ЗАХ) на карточке и на терминале к данным, представляемым карточкой (или терминалом) и к некоторому случайному числу, представляемому терминалом (или карточкой), когда начинается сеанс; код аутентификации сообщения (КАС), который получается в результате, подписывается карточкой (или терминалом) с помощью алгоритма цифровой подписи (АлЦП) и с использованием секретного ключа, содержащегося в карточке (или терминале); терминал (или карточка) верифицирует подпись, пользуясь открытым ключом. Для получения КАС и/или формирования подписей также можно использовать симметричный криптографический алгоритм, такой как стандарт шифрования данных (СШД).

В одном варианте реализации изобретения этап получения КАС является общим в обоих направлениях аутентификации и имеет отношение ко всем данным сеанса. При использовании симметричной криптографии сертификат, аутентифицирующий карточку, и сертификат, аутентифицирующий терминал, получают за один этап шифрования КАС, а соответствующие сертификаты карточки и терминала получают из них с помощью элементарной операции, такой как извлечение некоторых предварительно определенных битов.

Пример 2.

В этом примере реализации изобретения данные памяти организованы в виде секторов, причем каждый сектор содержит четыре поля:

1. данные;

2. идентификатор (ключа доступа, разрешающего выбор сектора);

3. релевантность: для определения того, какой сектор является релевантным (подходя-

щим), если два сегмента имеют одинаковый идентификатор; и

4. контроль: для верификации того, что три предшествующих поля не искажены (например, путем проведения контроля на четность).

Сектор обозначают его идентификатором, причем это обозначение заменяет обозначение адреса. Процедура записи в сектор имеет идентификатор в качестве параметра вместе с данными для связи с таким идентификатором. Процедура считывания сектора имеет идентификатор в качестве своего параметра, и она возвращает данные, которые были связаны с идентификатором в последнем случае проведения записи с использованием такого идентификатора (или надлежащее указание, если идентификатор ранее никогда не использовался). Иными словами, вместо индексируемого доступа реализуется ассоциативный тип доступа.

Во время процедуры считывания сектора карточка осуществляет поиск секторов, имеющих идентификаторы, содержащие запрашиваемое значение, и являющиеся неискаженными (что определяется полем контроля). Когда множество секторов удовлетворяет этим двум критериям, конкретный сектор сохраняется на основании поля релевантности.

При записи сектора карточка записывает в доступном секторе следующее: запрашиваемые данные, идентификатор, поле релевантности, такое, что во время процедуры считывания этот сектор будет наиболее релевантным из неискаженных секторов, обладающих этим идентификатором, и поле контроля, согласующее три предыдущих поля (иными словами, управление записью таково, что последующее считывание может происходить надлежащим образом).

За процедурой записи предпочтительно следует стирание сектора, который оказался нерелевантным (неподходящим), путем записи нового сектора, что делает новый сектор доступным.

Предпочтительно также предусмотреть систему, которая обеспечивает чистку памяти (сбор ненужной информации ("мусора")), т.е. систему для восстановления секторов, которые не используются либо потому, что они искажены, либо потому, что они не релевантны.

Предпочтительно предусмотреть систему, которая распределяет износ, являющийся результатом записи, гарантируя, что не всегда используются одни и те же секторы, например - путем выбора сектора случайным образом среди секторов, которые доступны.

Предпочтительный в общем случае вариант процедуры поиска сектора заключается в том, что используют преимущество этапа поиска для стирания секторов, которые, как обнаружилось, искажены, и/или секторов, которые не являются наиболее релевантными, обновляя таким образом три сектора (что занимает время в процессе конкретного считывания, способствуя ускорению последующих считываний и записей). Предпочтительно, перед стиранием сектора, который, как обнаружилось, оказался неискаженным, но нерелевантным, снова проводят запись в релевантный сектор, поскольку запись в него могла быть произведена не надлежащим образом.

Рабочая зона памяти равна числу доступных секторов минус один сектор, который должен оставаться стертым. Все секторы (включая стертый сектор) динамически распределяются в пределах памяти.

Если данные приходится структурировать в файлах, например, в случае применения стандарта 7816-4 Международной организации по стандартизации и Международной электротехнической комиссии (ИСО/МЭК), то идентификатор сектора подразделяют на два субполя: идентификатора поля и идентификатора для сектора в пределах файла.

Неограничительная реализация операций считывания-записи с использованием этой конкретной структуры сектора приведена ниже.

Далее следует описание (неограничительной) реализации операций считывания-записи с использованием этой конкретной структуры сектора.

- Поле контроля содержит выраженное в двоичном коде число нулевых битов в остальных трех полях; было обнаружено, что если имеет место некоторая проблема, такая как прерванная запись или стирание, которая изменяет любое число битов в секторе, в одном и том же направлении, то контроль значения поля контроля всегда может способствовать обнаружению возникновения этой проблемы.

- Поле релевантности представляет собой целое число в диапазоне 0-3, закодированное на 2-х битах.

- Процедура считывания обеспечивает последовательное считывание всех секторов до тех пор, пока не обнаружится первый сектор, который обладает искомым идентификатором и который не искажен. Если сектор не обнаружен, то процедура оканчивается сообщением "сектор не обнаружен". Если первый такой сектор обнаружен, то его положение запоминают вместе с его данными и его релевантностью p. Поиск продолжается. Если обнаружен второй сектор, который обладает искомым идентификатором и который не искажен, то проверяют, является ли его релевантность q остатком от целочисленного деления числа p+1 на 3, если его релевантность является упомянутым остатком, то второй сектор повторно записывают, первый сектор стирают и возвращают данные из второго сектора; в противном случае, повторно записывают первый сектор, стирают второй сектор и возвращают данные из первого сектора. Если второй сектор не обнаружен и если релевантность первого сектора равна 3 (p=3), то этот сектор стирают и выдают сообщение "сектор не обна-

ружен"; в противном случае, возвращаемые данные поступают из первого обнаруженного сектора.

Процедура записи начинается аналогично вышеописанной процедуре считывания. Если обнаружен ранее запомненный сектор, который должен быть возвращен процедурой считывания для заданного идентификатора, то положение этого сектора сохраняют с его релевантностью p (которая равна 0, 1 или 2); если такой сектор не обнаружен, то выбирают свободный сектор (с использованием процедуры, описанной ниже) и записывают в упомянутый сектор поля идентификатора, данных, релевантности p=3 и контроля, а положение и релевантность упомянутого сектора сохраняют. В обоих случаях процедура продолжается выбором свободного сектора (с использованием процедуры, описанной ниже). В этот сектор записывают поля идентификатора, данных, релевантности (вычисляемой как остаток от целочисленного деления числа p+1 на 3) и контроля. Затем ранее запомненный сектор, если он есть, стирают.

- Для поиска свободного сектора число n обнаруженных свободных сторон инициализируют при нуле. Проводят последовательную проверку секторов. Для каждого сектора, который не является пустым и который искажен, осуществляют стирание сектора, так что он становится пустым (внося таким образом вклад в вышеупомянутую чистку памяти); если сектор не искажен и если его релевантность не соответствует p=3, то в еще не просканированной (не просмотренной) зоне осуществляют поиск другого неискаженного сектора, имеющего тот же идентификатор, и если его обнаруживают, то стирают нерелевантный сектор, действуя так же, как при считывании; если в конце этого процесса сектор оказывается пустым, то дают приращение числу n обнаруженных свободных секторов и извлекают случайное целое число в диапазоне от 0 до n-1; если это целое число равно 0, то положение пустого сектора запоминают. Если просканированы все секторы, все непустые секторы не искажены, никакие два сектора не имеют одинаковый идентификатор, то число n пустых секторов известно, и один из них запоминают в качестве случайного выбора, сделанного равновероятным образом. Если свободный сектор не обнаружен, то процедуру записи прерывают.

Ниже описывается способ, с помощью которого карточка может управлять сеансами неделимых изменений, используя такую конкретную структуру сектора.

Для запоминания неделимых изменений карточка имеет N стертых секторов, доступных в энергонезависимой памяти (где N соответствует числу неделимых изменений, внесение которых может потребоваться во время одного сеанса). Кроме того, карточка управляет зоной энергонезависимой памяти (не входящей в секторы), которая предназначена для управления сеансом и которую называют "описателем сеанса".

Эта реализация не имеет аутентификации, специфичной для сеанса.

Описатель сеанса определяется на трех полях:

- списка ссылок на неделимые секторы (ССНС);

- контрольного значения при создании списка ссылок на неделимые секторы (КЗССНС); и

- контрольного значения, учитывающего список ссылок на неделимые секторы (КЗУССНС), для определения того, завершен ли сеанс.

Этап 0: инициализация: перед первым доступом к данным после ближайшего по времени прерывания работы карточки, например, после сброса в исходное состояние, карточка должна гарантировать, что описатель сеанса стерт. Необходимо учесть несколько случаев в зависимости от состояния описателя сеанса:

- он полностью стерт: карточка оставляет его неизменным;

- он не полностью стерт, а КЗУССНС верен: карточка осуществляет поиск всех секторов, ставших устаревшими, и стирает их (при необходимости), заменяя теми, которые записаны (из тех, на которые есть ссылки в списке), а затем стирает описатель сеанса;

- он не полностью стерт, КЗУССНС стерт или неверен, а КЗССНС верен: карточка стирает секторы, заданные в ССНС, а затем стирает описатель сеанса; или

он не полностью стерт, КЗУССНС стерт или неверен и КЗССНС стерт или неверен: карточка стирает описатель сеанса.

Этап 1: начало сеанса: карточка осуществляет поиск N стертых секторов, а затем записывает список ссылок на них в КЗССНС в описателе сеанса (предполагается, что он был стерт).

Этап 2: протекание сеанса: карточка принимает команды. Когда одна из них вызывает одно или несколько изменений, секторы, используемые для записи этих изменений, являются секторами, записанными в ССНС, а их количество в сумме составляет до N измененных секторов.

Этап 3: завершение сеанса: чтобы завершить сеанс, карточка осуществляет запись КЗУССНС, который гарантирует учет ССНС и его КЗССНС. После этого она осуществляет поиск всех секторов, которые стали устаревшими, и стирает их, заменяя теми, которые записаны (из тех, на которые есть ссылки в списке). Затем она стирает описатель сеанса.

Кроме того, поскольку именно карточка управляет ратификацией, то управление сеансом включает следующие модификации.

Этап 0: инициализация: в случае, когда описатель сеанса не полностью стерт, а

КЗУССНС верен, карточка осуществляет поиск всех секторов, ставших устаревшими, и стирает их, заменяя теми, которые записаны (из тех, на которые есть ссылки в списке), но не стирает описатель сеанса.

Этап 1: начало сеанса: карточка записывает в энергозависимой памяти, что сеанс начался. Если описатель сеанса не является пустым, то карточка указывает, что предыдущий сеанс не ратифицирован, и путем анализа ССНС может даже указать, какие элементы данных не ратифицированы. В любом случае, она не изменяет описатель сеанса.

Этап 2: протекание сеанса: во время первой команды с неделимыми изменениями, карточка стирает описатель сеанса и, при необходимости, осуществляет поиск N стертых секторов, а затем записывает ССНС и его КЗУССНС.

Этап 3: завершение сеанса: карточка записывает в энергозависимой памяти, что начавшихся сеансов нет. Всякий раз, когда это случается, она не стирает описатель сеанса.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ изменения содержимого энергонезависимой памяти карточки с микросхемой, в частности бесконтактной карточки, согласно которому карточку временно подключают к терминалу во время выполнения транзакции, в частности транзакции дистанционного заказа билетов, причем транзакция включает в себя подачу из терминала в карточку множества команд изменения, каждая из которых содержит, по меньшей мере, одну операцию записи в память карточки соответствующего элемента данных, обозначенного упомянутой командой изменения, при этом различные элементы данных, записываемые таким образом, являются взаимозависимыми, отличающийся тем, что карточкой

а) после соответствующего приема соответствующих команд из терминала изменяют содержимое памяти карточки путем временной записи в памяти карточки каждого из взаимозависимых элементов данных информации без потери предыдущих значений, соответствующих упомянутым элементам, а затем

б) завершают изменения содержимого памяти ячейки либо подтверждением, либо отклонением всех упомянутых изменений, так что при последующих операциях команды, выполняемые на этапе а), либо будут все учтены, либо все будут безрезультатными.

2. Способ по п.1, отличающийся тем, что

в случае подтверждения на этапе б) флаг, подтверждающий надлежащее выполнение, записывают в памяти карточки, а

когда карточка впоследствии принимает команду, требующую осуществить считывание или изменение, по меньшей мере, одного из элементов данных, записанных на этапе а), или соответствующего ему значения, карточкой начинают проверять состояние флага, и если он не записан, карточкой игнорируют или удаляют записи, сделанные ранее на этапе а), и выполняют команду на основе упомянутых предыдущих значений, соответствующих элементам данных.

3. Способ по п.2, отличающийся тем, что при проверке карточкой состояния флага и обнаружении того, что он записан, карточкой выполняют операции копирования условных записей, сделанных на этапе а).

4. Способ по одному из пп.1 или 2, отличающийся тем, что карточка выполнена с возможностью работы в двух режимах, а именно,

во внутрисеансовом режиме, при котором записи вносят путем выполнения этапов а) и б), и

во внесеансовом режиме, при котором внесение записей не подтверждают для всех этапов а) и б).

5. Способ по любому одному из пп.1-4, отличающийся тем, что выполняют функцию аутентификации в сочетании с функцией завершения этапа б), в случае безуспешной аутентификации на этапе б) выполняют отклонение всех упомянутых изменений.

6. Способ по п.5, отличающийся тем, что упомянутую аутентификацию выполняют карточкой, аутентифицирующей терминал и/или данные, которыми терминал обменивается с карточкой, причем карточкой контролируют криптографический сертификат, создаваемый терминалом и передаваемый в карточку, и подтверждают изменения, внесенные на этапе б), только в случае, если сертификат признан правильным.

7. Способ по п.6, отличающийся тем, что карточка выполнена с возможностью работы в двух режимах, а именно,

во внутрисеансовом режиме, при котором записи вносят путем выполнения этапов а) и б), и

во внесеансовом режиме, при котором внесение записей не подтверждают для всех этапов а) и б),

и тем, что при приеме карточкой из терминала команды изменения содержимого памяти, включая верификацию криптографического сертификата, упомянутую верификацию выполняют при приеме команды во внесеансовом режиме и не выполняют при приеме команды во внутрисеансовом режиме.

8. Способ по п.5, отличающийся тем, что упомянутую функцию аутентификации выполняют терминалом, аутентифицирующим карточку и/или данные, которыми обмениваются терминал и карточка, причем карточкой создают и передают криптографический сертификат условным образом в терминал только при подтверждении изменений на этапе б).

9. Способ по п.1, отличающийся тем, что карточка выполнена с возможностью работы в двух режимах, а именно,

во внутрисеансовом режиме, при котором записи вносят путем выполнения этапов а) и б), и

во внесеансовом режиме, при котором внесение записей не подтверждают для всех этапов а) и б),

и тем, что при приеме карточкой из терминала команды изменения содержимого памяти, включая создание криптографического сертификата, упомянутое создание криптографического ключа выполняют при приеме команды во внесеансовом режиме и его не выполняют при приеме команды во внутрисеансовом режиме.

10. Способ по одному из пп.1 или 2, отличающийся тем, что при приеме карточкой из терминала команды изменения содержимого памяти, включая создание множества криптографических сертификатов, упомянутые сертификаты запоминают на этапе б), а затем передают вместе в терминал, при подтверждении упомянутых изменений на этапе б).

11. Способ по п.1, отличающийся тем, что, карточка выполнена с возможностью работы в двух режимах, а именно,

во внутрисеансовом режиме, при котором записи вносят путем выполнения этапов а) и б), и

во внесеансовом режиме, при котором внесение записей не подтверждают для всех этапов а) и б),

и тем, что по меньшей мере, некоторые из команд, которые могут быть выполнены на этапе б), включают в себя необязательный запрещающий атрибут, и если карточкой выполняют упомянутую команду во внутрисеансовом режиме на этапе б), то изменения, вносимые по упомянутой команде, оставляют в силе независимо от результата этапа б).

12. Способ по одному из пп.1 или 2, отличающийся тем, что после этапа б) и при подтверждении упомянутых изменений

г) терминалом выполняют его функцию после упомянутого подтверждения карточкой и

д) при выполнении упомянутой функции надлежащим образом терминалом информацию о ратификации записывают в карточку для последующего доступа путем считывания.

13. Способ по п.12, отличающийся тем, что команда записи на этапе д) является неявной командой, при этом любую команду, принимаемую карточкой после этапа б), интерпретируют как команду записи информации о ратификации в карточку.