

TRANSACTION SYSTEM

FIELD OF THE INVENTION

The present invention relates to a transaction system and
5 a transaction terminal, adapted to save transaction data
and of particular but by no means exclusive application
with offline transactions.

BACKGROUND OF THE INVENTION

10 An advantage of using a customer token such as a smart
card (e.g. a personal transaction or identification card
with an integrated circuit chip (or ICC) or other form of
memory) in transaction systems lies in the security that
can be provided by means of the ICC. Owing to this
15 security, some existing smart cards are used to store
monetary value in electronic form in what has become known
as 'electronic purse' systems. In such systems, a
transaction terminal is used to deduct monetary value from
the electronic purse in the smart card; this monetary
20 value is stored in the terminal in an offline mode. That
is, the terminal does not have to connect to a host system
in order to authorize the payment, since the transferred
monetary value constitutes full payment for the purchased
goods or services. The value deducted from the smart card
25 is stored in the terminal and only subsequently sent to a
host computer as part of a batch of such transactions.
Such an arrangement allows the terminal to connect to the
host system infrequently (perhaps once a day at closing
time), instead of at the time of every transaction.

30 Another example of an existing offline application of this
kind involves the use of a smart card to store 'rewards'
or bonus points in a customer loyalty scheme. The bonus
points are stored as a stored value in the smart card,
35 which can be deducted by an authorised terminal as part of
or in lieu of a cash payment. Such offline transactions
are captured in the terminal and - again - transmitted to

a host system in batches to reduce cost of data communications incurred when contacting the host system.

5 A typical existing technique for making such offline systems secure involves authenticating the ICC in the smart card. This is done by means of cryptographic systems that authenticate the ICC card and therefore verify the card's authenticity before accepting the card for use in a transaction. Such existing authentication
10 schemes require that the terminal store a secret value, known as a 'Key', that is used by the terminal to verify the authenticity of the smart card. Typically this authentication is performed by a so-called 'mutual authentication' process.

15 To effect such a mutual authentication process, it is assumed that the ICC in the smart card contains a secret Key and the terminal attempts to verify this algorithmically by exchanging random values with the smart
20 card. The process is known as mutual authentication because the card will similarly attempt to verify whether the terminal has a corresponding secret Key, that is, whether the terminal is an authentic terminal.

25 This process of mutual authentication is effective only if the terminal's secret Key is kept secret; unauthorized access to the secret Key must be prevented. This requires that the terminal be tamper-resistant and generally prevents unauthorized access to that Key. To achieve
30 this, it is a common industry practice to store the secret Key or secret Keys in a tamper-resistant device (such as an ICC card housed in the terminal) that is programmed to function as a so-called Secure Access Module (or SAM). The SAM is physically secure and designed not to release
35 the secret value or Key. The SAM is typically installed in a dedicated slot in the terminal into which the SAM can be inserted and from which it can be retrieved. A

terminal may have multiple SAM slots to allow multiple SAMs to be installed.

As discussed above, therefore, in such existing systems
5 offline transactions captured in the terminal must be transmitted to the host system so that the system can account for the transaction. In the case of an electronic purse system, the system must pay the merchant the amount collected in electronic form, while a customer loyalty
10 system must account for the amount of rewards deducted from or added to the smart card. However, if the host system is unable to receive the offline transaction information, the transaction cannot be accounted for and either the merchant is not paid (in the electronic purse
15 system) or, in the case of the customer loyalty system, a customer's use of rewards in the offline mode can remain unrecorded by a host system which consequently records those rewards as unused and available to the customer.

20 Hence, loss of offline transactions stored in terminals of existing offline transaction systems can be a serious problem. Sometimes the loss of such transactions can be corrected by keying in the transaction data from receipt journals or copies of the receipts printed at the terminal
25 at the time of transaction. This is laborious, error prone and difficult to check for authenticity.

SUMMARY OF THE INVENTION

It is an object of the present invention, therefore, to
30 provide a transaction system that takes advantage of the memory storage of the SAM to at least ameliorate the aforementioned problems.

According to a first aspect of the present invention,
35 therefore, there is provided a transaction system comprising:

a host computer server; and

at least one electronic transaction terminal for receiving transaction data pertaining to customer transactions and for transmitting said transaction data to said server over a computer network, said transaction
5 terminal including a memory, an interface for establishing data communication with a data storage device for storing at least one digital security key, and a customer token interface for establishing data communication with a customer token;

10 wherein said security key is adapted to facilitate authentication of at least one said transaction terminal and said customer token, and is accessible by said transaction terminal when said data storage device is in data communication with said transaction terminal, said
15 data storage device is removably engagable with said transaction terminal and said transaction terminal is configured to write said transaction data to said memory for subsequent retrieval and to said data storage device for subsequent retrieval by said transaction terminal or
20 another like transaction terminal.

Preferably the data storage device comprises a secure access module or SAM.

25 Thus, a further copy of the transaction data is stored in the transaction terminal's memory. This allows the transaction terminal to transmit the transaction data to the server after retrieving the data either from the data storage device or from the memory of the transaction
30 terminal. Typically, the copy of the transaction data saved in the terminal's memory would be treated as the principal copy of the transaction data, with the copy stored in the data storage device acting as a backup.

35 Thus, the invention enables transaction data (and most especially transaction data from offline transactions) to be stored redundantly and cost effectively by taking

advantage of an existing data storage or memory device such that the transaction data can still be retrieved from the memory device in the event that the terminal becomes faulty.

5

The transaction data is stored to and retrieved from the SAM independently of the security functions that a SAM is typically used for, without compromising the SAM's existing security functions (such as protection of the cryptographic key or keys stored in the SAM).

10

Preferably the transaction terminal includes a housing (such as a card slot or a smart card acceptor) for receiving said data storage device, wherein said data storage device can be connected to said interface when in said housing.

15

The transaction terminal may be configured to delete said transaction data from said data storage device after said transaction data has been successfully transmitted to said server by said transaction terminal.

20

Hence, once the transaction data has been successfully transferred to the host server, it need no longer be saved on the data storage device so it can be deleted from that device. The data storage in the data storage device is then available for subsequent transaction data.

25

In one embodiment, the transaction terminal is configured to delete said transaction data from said data storage device after said transaction data has been successfully transmitted from said data storage device to said transaction terminal. In this embodiment also, therefore, data storage in the data storage device is made available for subsequent transaction data.

30

35

Preferably the data storage device is configured to write

new transaction data over old transaction data when said data storage device is full of transaction data, wherein the oldest transaction data is overwritten first.

5 Preferably the transaction terminal is configured to write to said data storage device only transaction data that pertains to offline transactions performed at the transaction terminal. Thus, in the event that the transaction terminal is faulty or for any other reason is
10 unable to transmit the offline transactions stored either in the terminal or on the data storage device (such as a SAM) to the host system, the data storage device can be removed from the terminal and plugged into another terminal, which can then retrieve transaction data stored
15 in the data storage device for transmission to the host.

In one embodiment, the interface is adapted to establish data communication with a plurality of removable data storage devices whereby a first data storage device can be
20 used to store said digital security key, and said transaction terminal is configured to write said transaction data to a second data storage device for subsequent retrieval by said transaction terminal or another like transaction terminal.

25 Preferably said transaction data is stored in said data storage device in a file containing a fixed amount of space, and said data storage device is configured such that when the number of transactions recorded in said file is at or above a threshold value, said data storage device
30 returns a status to said transaction terminal prompting said transaction terminal to upload any transaction data stored in said transaction terminal to said server to thereby clear offline transaction data stored in said
35 transaction terminal and any redundant copies of said transaction data stored in said data storage device, so that said transaction terminal and said data storage

device can hold more transaction data without losing any transaction data previously stored in said data storage device.

5 According to a second aspect of the invention there is provided an electronic transaction terminal for receiving transaction data pertaining to customer transactions and for transmitting said transaction data to a host server over a computer network, said terminal comprising:

10 a memory; an interface for establishing data communication with a removable data storage device for storing at least one digital security key; and

a customer token interface for establishing data communication with a customer token;

15 wherein said security key is adapted to facilitate authentication of at least one of said transaction terminal and said customer token, and is accessible by said transaction terminal when said data storage device is in data communication with said transaction terminal, said data storage device is
20 removably engagable with said transaction terminal and said transaction terminal is configured to write said transaction data to said memory for subsequent retrieval and to said data storage device, whereby said transaction
25 data in said data storage device can subsequently be retrieved by said transaction terminal or another like transaction terminal.

Preferably said transaction terminal is configured to
30 write said transaction data to said memory and said data storage device, as respective transaction logs. Preferably the digital security key is one of a plurality of digital security keys, more preferably for authenticating customer tokens.

35

The interface may be adapted to establish data communication with a plurality of removable data storage

devices whereby a first data storage device can be used by said transaction terminal to store said digital security key, and said transaction terminal is configured to write said transaction data to both a transaction log in said transaction terminal and to a second data storage device for subsequent retrieval by said transaction terminal or another like transaction terminal.

According to a third aspect of the invention there is provided a method of saving transaction data before transfer to a host server in a transaction system comprising:

writing said transaction data by means of a transaction terminal to a memory provided in said transaction terminal and to a data storage device that is removably engaged with said transaction terminal has a digital security key for facilitating authentication of at least one of said transaction terminal and a customer token, and is in data communication with said transaction terminal;

whereby said transaction data in said data storage device can subsequently be retrieved by said transaction terminal or another like transaction terminal.

The method may include deleting said transaction data from said data storage device if said transaction data has been successfully transmitted by said transaction terminal to said host server.

The method may include writing to said data storage device only transaction data that pertains to offline transactions performed at said transaction terminal.

BRIEF DESCRIPTION OF THE DRAWING

In order that the invention may be more clearly ascertained, preferred embodiments will now be described, by way of example, with reference to the accompanying

drawings, in which:

Figure 1 is a schematic representation of a transaction system according to a preferred embodiment of the present invention;

5 Figure 2 is a schematic representation of a payment terminal with SAM and customer token of the system of figure 1;

10 Figure 3A is a flow chart illustrating the process for storing transaction data from a payment terminal of the system of figure 1; and

 Figure 3B is a flow chart illustrating the process of retrieval of transaction data from a Secure Access Module to a payment terminal of the system of figure 1.

15

DETAILED DESCRIPTION OF THE DRAWING

A transaction system according to a preferred embodiment of the present invention is shown schematically in figure 1 at 10. The system 10 includes a host server 12 and a
20 plurality of transaction terminals in the form of payment terminals 14, connected to the host server 12 by means of a computer network 16 (such as the internet). Each payment terminal 14 is designed to permit a customer to conduct a transaction by means of a payment card, a
25 loyalty card or the like.

Referring to figure 2, each payment terminal 14 has a processor 17 connected to non-volatile memory (NVM) 18 and random-access memory (RAM) 19, a keyboard 20, a receipt
30 printer 22, and a SAM acceptor 24. The SAM acceptor 24, providing the payment terminal's interface with a SAM, is in the form of a smart card acceptor such as one conforming to International Standards 7816 Parts 1 to 3 (for example, one of the type commonly used for subscriber
35 identity modules (sims) in mobile phones).

The payment terminal 14 also includes a card acceptor 26 such as one conforming to International Standards 7816 Parts 1 to 3. The card acceptor 26 comprises, for example, one of the type found on Electronic Draft Capture
5 terminals used in credit card payment systems conforming to the Europay-Mastercard-Visa (EMV) standard specifications. The card acceptor 26 could alternatively be adapted to accept 'contactless' ICC cards using radio frequency (RF) transmission techniques (such as those
10 complying with International Standards 1444) for communication between the terminal and such ICC cards.

The payment terminal 14 includes (when in use) a SAM 28, installed in the SAM acceptor 24. The SAM 28 has an
15 integrated circuit chip 30 with built-in anti-tampering security features to protect the digital contents of the chip; the chip 30 comprises a processing unit 32, read-only memory (ROM) 34 with application software coded to form an operating system on the chip, volatile RAM 36,
20 non-volatile memory (NVM) 38 (such as Electrically Erasable and Programmable Read-only Memory (EEPROM)) containing software program coded to perform the functions described below as well as associated data and an I/O unit
40.

25 The system also includes a plurality of customer tokens, each in the form of a customer smart card 42 containing an integrated circuit chip 44. ICC 44 comprises a processor unit 46 connected to a ROM area 48 containing a software
30 application coded to perform payment functions in the chip 44, a RAM area 50, and an input/output unit 52 (for interfacing with the card acceptor 26 of the payment terminal 14).

35 When a customer smart card 42 is used at the payment terminal 14 to make a payment or some other type of customer transaction, a transaction record is generated

and stored in the NVM 18 of the payment terminal 14 for subsequent transmission to the host computer server 12 for settlement, reconciliation, etc.

5 This transaction record is also sent to the SAM 28 for logging. Implemented in the SAM 28 is a software program stored in the NVM 38 of SAM 28; this software program is configured to accept and log such transactions. The SAM 28 receives the transaction data from the payment terminal 14, and invokes a function that: 1) looks for the next available memory location in the NVM 38 of SAM 28, 2) logs the transaction in that location, and 3) updates an index to indicate the location of the memory area available for the next transaction.

15

This is repeated for every transaction, until the SAM 28 runs out of storage area in its NVM 38. When this occurs, the index is updated to point to the memory location at the start of the transaction logging area in the SAM's NVM 38. This means that the next transaction will be logged in at the start of the transaction logging area in this NVM 38, thus over-writing the transaction record previously logged there. The number of transactions that SAM 28 can cater for (that is, the number of transactions that can be logged in the SAM before the oldest transaction in the SAM is overwritten) depends on the number of transactions that is likely to occur before the terminal 14 can transmit its batch of transaction data to the host. Each batch is typically identified by a batch number, which may be assigned in a variety of ways (such as by assigning a sequential number incremented for each new batch, or according to the date and time of the first transaction). Each time a batch is uploaded successfully, a new batch is started, and the terminal 14 issues a command to the SAM 28 to mark the start of a new batch by updating in the SAM the new batch number.

If the payment terminal 14 fails and the transaction logged in its NVM is lost, the SAM 28 can be removed from the payment terminal 14 and inserted into another payment terminal 14. This second payment terminal 14 can be
5 programmed to make requests to the SAM 28 to retrieve the transaction data stored in the NVM 38 of the SAM 28 from the SAM and send the data to the host server 12 in place of the original transactions in the damaged payment terminal 14.

10

Optionally, each time the payment terminal 14 starts a new batch of transactions, the change (i.e. from a previous batch of transaction to the new batch) is also logged in the SAM 28. This allows the payment terminal 14 to
15 recover transactions from the SAM starting from the start of the last batch logged in the SAM 28, and allows the SAM to reuse the NVM 38 space in the SAM 28 used by transactions from earlier batches which have been successfully uploaded and are no longer needed to be
20 backed-up on the SAM 28.

Figure 3A is a flow chart that illustrates the process (beginning at 'A') of storing a transaction into the SAM 28 in the normal course of processing a transaction
25 ('Txn') in the terminal 14 when the terminal 14 is operating offline (i.e. without connection to the host server 12). In this flow chart, Idx refers to the identity number of a transaction x, StartIdx is the first Idx of a particular batch of transactions and EndIdx is
30 the last Idx of a particular batch of transactions.

At set intervals determined by any of a number of conditions (such as the number of transactions captured in the terminal or the time of day), the terminal 14 proceeds
35 to send transactions (that is, upload transactions)_to the host server 12. At the end of a successful upload process, the terminal sends to the SAM a 'Start new batch'

command to demarcate transactions captured in SAM before the upload and those to be captured after upload. This 'Start new batch' process is also illustrated in Figure 3A.

5

In greater detail, the sequence of events illustrated in figure 1 is as follows.

When a new SAM is first inserted into the terminal 14, the terminal 14 starts a new batch in SAM 28 (at step 66) as part of an initialisation process. The following description assumes that first-time "start new batch" steps 66, 67, 68 and 69 have been completed, and therefore commences at 'A' (step 53) with the more generic case, in which the SAM 28 is used in part of a transaction process for logging a transaction:

Step 53: User presents smart card 42 to terminal 14;

Step 54: The transaction (such as a payment transaction or a loyalty points or coupon transaction) takes place,

20 possibly with data updated in the card 42;

Step 55: Transaction data logged to transaction log ('TLOG') in the terminal 14;

Step 56: Terminal 14 sends transaction data to the SAM 28 for updating the transaction log ('SLOG') in SAM 28;

25 Step 57: The Logging Application in SAM 28 has predefined amount of memory for storing SLOG. This memory is divided into space for a maximum number of transaction records denoted by MaxIdx. SAM application also has a number of other data storage elements (variables in programming jargon) named and defined as follows:

- Idx is the location of the last transaction written into the SLOG; initial value of Idx in a new SAM is 0.
- StartIdx is the location of the 1st transaction record written into the SLOG for the current batch.
- 35 • EndIdx is the location of the last transaction record written into the SLOG for the batch.
- Batch# identifies a batch.

- Batch Directory is a location in the SAM containing a list of Batch#s, and each batch identified by a Batch# has a corresponding StartIdx and EndIdx indicating the position of the first and last transaction record for that batch.

Thus, in this step, check if Idx of current batch is pointing at MaxIdx (i.e. the last record in SLOG);

Step 58: If YES, then point Idx back to the top of SLOG, that is, set Idx = 1 and continue at step 60;

Step 59: If NO (i.e. Idx is not pointing to the last record in SLOG), increment Idx to point to the next record location and continue at step 60;

Step 60: Write the current transaction into the SLOG record location indicated by Idx. Set EndIdx to Idx, to indicate that the last transaction of this batch is here;

Step 61: Check whether the last transaction of the current batch has overlapped into another batch. This is determined by checking whether EndIdx of the current batch is equal to StartIdx of any other batch (a "second batch") in the Batch Directory;

Step 62: If YES, the second batch is no longer valid and its Batch Directory entry is removed from the Batch Directory, and procedure continues at step 63; otherwise go to step 63;

Step 63: Terminal 14 checks to determine whether any upload is to take place (according to parameters defining the condition for upload);

Step 64: If YES (and conditions for upload are met), terminal initiates upload then continue at step 65;

otherwise return to 'A' (step 53);

Step 65: Check whether upload was successful;

Step 66: If YES (but also if this is the first time the SAM 28 is inserted in the Terminal), terminal 14 instructs SAM 28 to start a new batch, then continues to step 67.

Otherwise, go to 'D' in figure 3B;

Step 67: SAM 28 updates the Batch Directory with new Batch entry, and sets both EndIdx and StartIdx of the new Batch

to Idx+1 (i.e. the Idx of the previous Batch or Idx of a new SAM inserted into the Terminal for the first time in which case the Idx is equal to 0) if Idx < MaxIdx, or sets StartIdx and EndIdx to 1 if Idx=MaxIdx;

5 Step 68: SAM 28 checks whether EndIdx has over-written the first record of an adjacent (other) batch by comparing the EndIdx of the new batch with StartIdx of other batch entries in the Batch Directory;

10 Step 69: If YES and another batch has a StartIdx equal to the current EndIdx value, that other batch entry is removed from the Batch Directory, as the data in the other batch is considered no longer valid as the records in that batch are being overwritten by records of the new batch, then return to step 53; otherwise, return to step 53.

15

Figure 3B is a flow chart illustrating the process that occurs according to the present embodiment if transaction uploading from a first payment terminal is found (at step 65 in figure 3A) to have failed. This necessitates the retrieval of transaction data from SAM 28 of the first payment terminal to a second payment terminal 14.

20

Typically the process shown in figure 3B would be used when it is found that the failure of the upload was due to a failure of the first payment terminal.

25

The failure of data communications between a particular terminal and the host server 12 is regarded as equivalent to terminal failure. In such cases, transferring the SAM to a second payment terminal for uploading the transaction data is a convenient way of by-passing the failure, without having to await the restoration of data communications between the first terminal and the host server 12.

30

35 It will be noted that, in the process shown in figure 3B, the user can select which Batch to upload from the SAM 28; specifically, the user is asked (at step 70) whether the

Batch to be uploaded is the last Batch (since one would usually attempt to restore a Batch that has not been uploaded from the terminal 14 essentially immediately) or not.

5

If not, the terminal 14 retrieves a list of Batches from the SAM and displays that list for user selection. In either case, therefore, the terminal 14 ultimately sends (at step 80) the selected Batch# to SAM 28 and - at step 10 90 - then sends a 'Get Txn' request to SAM 28. At step 100, a counter N is set to StartIdx after which, in sub-process 110, each transaction N in the instant Batch is uploaded from the SAM 28 to the terminal 14, and N is incremented until N equals EndIdx.

15

It should also be noted that, because the terminal 14 marks the start of each batch of transactions logged in the SAM with a Batch Id (see figure 3A), SAM 28 can be configured to track multiple batches, depending on the 20 amount of memory/data storage is available for recording transactions in the SAM 28. Thus, during upload from SAM 28, the terminal 14 may be programmed to allow the user to select a batch that was the subject of a previous upload attempt from the SAM (following some still earlier upload 25 failure from the terminal 14), and re-attempt that earlier failed upload from the SAM 28.

Modifications within the scope of the invention may be readily effected by those skilled in the art. It is to be 30 understood, therefore, that this invention is not limited to the particular embodiments described by way of example hereinabove.

CLAIMS:

1. A transaction system comprising:

a host computer server; and

at least one electronic transaction terminal for
5 receiving transaction data pertaining to customer
transactions and for transmitting said transaction data to
said server over a computer network, said transaction
terminal including a memory, an interface for establishing
data communication with a data storage device for storing
10 at least one digital security key, and a customer token
interface for establishing data communication with a
customer token;

wherein said security key is adapted to
facilitate authentication of at least one of said
15 transaction terminal and said customer token, and is
accessible by said transaction terminal when said data
storage device is in data communication with said
transaction terminal, said data storage device is
removably engagable with said transaction terminal and
20 said transaction terminal is configured to write said
transaction data to said memory for subsequent retrieval
and to said data storage device for subsequent retrieval
by said transaction terminal or another like transaction
terminal.

25

2. A system as claimed in claim 1, wherein said data
storage device comprises a secure access module.

3. A system as claimed in claim 1, wherein said
30 transaction terminal includes a housing for receiving said
data storage device, wherein said data storage device can
be connected to said interface when in said housing.

4. A system as claimed in claim 1, wherein said
35 transaction terminal is configured to delete said
transaction data from said data storage device after said
transaction data has been successfully transmitted to said

server.

5. A system as claimed in claim 1, wherein said transaction terminal is configured to delete said transaction data from said data storage device after said transaction data has been successfully transmitted from said data storage device to said transaction terminal.

6. A system as claimed in claim 1, wherein said data storage device is configured to write new transaction data over old transaction data when said data storage device is full of transaction data, wherein the oldest transaction data is overwritten first.

7. A system as claimed in claim 1, wherein said transaction terminal is configured to write to said data storage device only transaction data that pertains to offline transactions performed at the transaction terminal.

8. A system as claimed in claim 1, wherein said interface is adapted to establish data communication with a plurality of removable data storage devices whereby a first data storage device can be used to store said digital security key, and said transaction terminal is configured to write said transaction data to a second data storage device for subsequent retrieval by said transaction terminal or another like transaction terminal.

9. A system as claimed in claim 1, wherein said transaction data is stored in said data storage device in a file containing a fixed amount of space, and said data storage device is configured such that when the number of transactions recorded in said file is at or above a threshold value, said data storage device returns a status to said transaction terminal prompting said transaction terminal to upload any transaction data stored in said

transaction terminal to said server to thereby clear transaction data stored in said transaction terminal and any redundant copies of said transaction data stored in said data storage device, so that said transaction
5 terminal and said data storage device can hold more transaction data without losing any transaction data previously stored in said data storage device.

10. An electronic transaction terminal for receiving
10 transaction data pertaining to customer transactions and for transmitting said transaction data to a host server over a computer network, said terminal comprising:
a memory;
an interface for establishing data communication
15 with a data storage device for storing at least one digital security key; and
a customer token interface for establishing data communication with a customer token;
wherein said security key is adapted to
20 facilitate authentication of at least one of said transaction terminal and said customer token, and is accessible by said transaction terminal when said data storage device is in data communication with said transaction terminal, said data storage device is
25 removably engagable with said transaction terminal and said transaction terminal is configured to write said transaction data to said memory for subsequent retrieval and to said data storage device, whereby said transaction data in said data storage device can subsequently be
30 retrieved by said transaction terminal or another like transaction terminal.

11. A transaction terminal as claimed in claim 10,
wherein said transaction terminal includes memory and is
35 configured to write said transaction data to said memory and said data storage device as respective transaction logs.

12. A transaction terminal as claimed in claim 10,
wherein said digital security key is one of a plurality of
digital security keys.

5

13. A transaction terminal as claimed in claim 10,
wherein said plurality of digital security keys are
adapted to authenticate customer tokens.

10 14. A transaction terminal as claimed in claim 10, wherein
said data storage device comprises a secure access module.

15 15. A transaction terminal as claimed in claim 10,
wherein said transaction terminal includes a housing for
receiving said data storage device, wherein said data
storage device can be connected to said interface when in
said housing.

20 16. A transaction terminal as claimed in claim 10,
wherein said interface is adapted to establish data
communication with a plurality of removable data storage
devices whereby a first data storage device can be used by
said transaction terminal to store said digital security
key, and said transaction terminal is configured to write
25 said transaction data to both a transaction log in said
transaction terminal and to a second data storage device
for subsequent retrieval by said transaction terminal or
another like transaction terminal.

30 17. A method of saving transaction data in a transaction
system against failure of uploading to a host server
comprising:

writing said transaction data by means of a
transaction terminal to a memory provided in said
35 transaction terminal and to a data storage device that is
removably engaged with said transaction terminal, has a
digital security key for facilitating authentication of at

least one of said transaction terminal and a customer token, and is in data communication with said transaction terminal;

5 whereby said transaction data in said data storage device can subsequently be retrieved by said transaction terminal or another like transaction terminal.

10 18. A method as claimed in claim 17, wherein said data storage device comprises a secure access module.

15 19. A method as claimed in claim 17, including writing to said data storage device only transaction data that pertains to offline transactions performed at said transaction terminal.

20 20. A method as claimed in claim 17, including deleting said transaction data from said data storage device if said transaction data has been successfully transmitted by said transaction terminal to said server.

25 21. A method as claimed in claim 17, including deleting said transaction data from said data storage device after said transaction data has been successfully transmitted from said data storage device to said transaction terminal.

30 22. A method as claimed in claim 17, further comprising writing new transaction data over old transaction data when said data storage device is full of transaction data, including writing over the oldest transaction data first.

35 23. A method as claimed in claim 17, including storing said transaction data in said data storage device in a file containing a fixed amount of space and, when the number of transactions recorded in said file is at or above a threshold value, said data storage device prompting said transaction terminal to upload any transaction data stored

in said transaction terminal to said server to thereby
clear transaction data stored in said transaction terminal
and any redundant copies of said transaction data stored in
said data storage device, so that said transaction terminal
5 and said data storage device can hold more transaction data
without losing any transaction data previously stored in
said data storage device.