Jun 22, 1999

APPLICATION NOTE 84

Add-Only Memory for Secure Storage of Monetary Equivalent Data

This document presents a scheme showing how Add-Only (memory) iButtons® can be used for applications requiring secure crediting, debiting, and portable storage of monetary equivalent data. Specifically, the document points out that Add-Only iButtons make an excellent replacement for magnetic stripe cards in said applications due to the advantages Add-Only iButtons have over magnetic stripe cards, such as low cost read/write mechanisms, being less prone to accidental erasure from magnetic fields, and being less prone to counterfeiting.

I. Introduction

The fare payment system used by the Bay Area Rapid Transit (BART) system in San Francisco is an example of an application in which monetary-equivalent data is read and written electronically. In this system, the user can obtain a transit ticket and deposit any desired amount of money into it from an automatic vending machine. The information is stored in the ticket magnetically in the form of encoded data written on a magnetic stripe. Each time the user travels from one place to another, the system deducts the fare from the amount represented by the magnetically encoded data, thus reducing the value of the ticket. When the value of the ticket is nearly exhausted, it can be restored to a high value by inserting it again into an automatic vending machine and depositing additional funds.

The BART system eliminates the need for handling money and making change at the point of entry to the transit system, thereby reducing labor costs and increasing efficiency. A similar advantage can be realized in many other circumstances where an electronically readable and alterable "token" can eliminate the costs and delays associated with money handling at the point of use. Such a token might therefore be used as a meal ticket on a college campus, as a ride ticket at an amusement park, or wherever tickets or tokens are now used to speed monetary payments and/or eliminate unnecessary labor.

The system described above suffers from three significant disadvantages:

1. Paper tickets with magnetic stripes deposited on them are subject to wrinkling or tearing, which can cause loss of the monetary equivalent data. Also, the magnetic stripes are subject to erasure by environmental magnetic fields, even if the paper carrier and magnetic material are physically intact.
2. Since magnetic recording is a read/write technology, it is possible for a technologically sophisticated person to read the contents of the magnetic stripe when the ticket has a large monetary value, use the ticket until the value is nearly gone, then rewrite the original data into the ticket to restore its original value. It is not necessary for the person to understand the encoding of the monetary data in order to do this. Therefore, the use of a read/write technology makes the tickets vulnerable to counterfeiting.
3. The magnetic recording technology requires uniform motion of the magnetic material across the read/write heads in order to read and write data reliably. This makes it necessary to use a relatively complex mechanical ticket-handling mechanism to read, debit, and rewrite the monetary equivalent data.

II. Add-Only iButton as an Alternative Technology for Monetary Tokens

Add-Only iButton provides a viable alternative technology for storage of monetary equivalent data which delivers the advantages described above but does not suffer from the disadvantages. The iButton is sealed in a durable stainless steel Microcan which protects it against environmental damage. Reading and writing data is accomplished with a

momentary contact to a simple electrical probe, requiring no sophisticated mechanical handling mechanisms. The add-only attribute of Add-Only iButton provides protection against counterfeiting, since the data in these memories can never be restored to its original value once it has been modified.

Add-Only iButton contains many bits of information, with each bit having either a 1 or a 0 value. Initially, all the bits in the memory are 1s. The read/write probe can read these bits, and it can also selectively change one or more of the bits to 0. Once a bit has been changed to a 0, it cannot be changed back to a 1. Writing a bit is therefore much like punching a hole in a meal ticket card. The electrically alterable bits are organized into memory pages having 256 bits each. In addition to these electrically alterable bits, each iButton also contains a unique 64-bit registration number which cannot be altered. No two iButtons ever have the same registration number. Finally, each page has a status register that can be read to determine which pages have been used up, and error detection circuitry (CRC) which allows the reader to determine if it has read the data correctly.

With this feature set, it is possible to design a system in which monetary equivalents can be added to or removed from the part many times before it must be replaced and which is highly resistant to counterfeiting. The basic principle of this system is described below.

III. Electronic Crediting and Debiting of Add-Only iButton

One possible technique which allows storage of credits and debits in Add-Only iButton is as follows. Monetary units are added by changing 1 bits to 0 bits starting from the least significant bit of each page and progressing toward the most significant bit. Monetary units are debited by changing 1 bits to 0 bits starting from the most significant bit of each page and progressing toward the least significant bit. As the memory is repeatedly debited and credited, the rows of 0 bits grow toward the middle of the page. When they meet, the page is marked as exhausted with the status byte and the process continues on the next page. (It is possible to ignore pages and treat the entire memory as a single page, but that would require the reading of the entire memory, increasing the time needed to complete a transaction. The electronic read/write process is more efficient when only a portion of the stored data needs to be read). With this technique, assuming the credit units and debit units have equal value, a 1024 bit memory could credit and debit 512 monetary units before it was used up. If credit units are taken to represent some multiple of the debit unit, then more debits are allowed. (For example, if each credit unit is the equivalent of three debit units, then a 1024 bit memory would allow 768 debits).

The problem with the simple system described above is that anyone with the necessary knowledge and equipment to read and write data in Add-Only iButton can easily increase the value by adding additional credit units. This is possible because there is a direct, straight-forward correspondence between a bit location and its value. If the bits were scrambled (permuted) in an apparently random manner, it would no longer be possible to determine how to add credit units to the memory. For example, if 15 bits on a page are still set to 1, only one of these bits will add a credit unit to the memory. Similarly, only one of the bits will add a debit unit to the memory. If any one of the other 13 bits were written to 0, it would appear out of sequence and would signify that the memory had been tampered with, thereby invalidating it. Therefore, if a person guesses which bit to write next, he has one chance in 15 of adding a credit unit, one chance in 15 of adding a debit unit (decreasing the value), and 13 chances in 15 of invalidating the memory and flagging it as having been subject to tampering. Although there is a chance of guessing correctly which bit to change, the laws of probability are stacked against this event. (This is the kind of statistical analysis that makes lotteries predictable).

The unique registration number in each iButton can be used to permute the bits in each part differently, so that one cannot determine by studying the data in one part how to add credit units to a different part. Many different techniques are possible to determine a unique bit permutation from the unique registration number supplied with each part. A few of these techniques are described below.

IV. Calculating Bit Permutations from Unique Registration Numbers

The number of different permutations of the 256 bits in each page is very large, approximately 10 to the power of 507. Only a minute fraction of these permutations can be enumerated with the unique registration number, since the registration number represents a range of 281 trillion unique numerical values, or about 10 to the power of 14. The

permutations that can be derived from the unique registration number are thus buried in the much larger population of possible permutations. 281 trillion is in fact an extremely large set of unique registration numbers that is sufficient for all practical purposes. The enormously larger number of different permutations greatly multiplies the task of deducing the permutation from the registration number. To select a permutation based on the registration number from this enormous population, the following method could be used.

1. Replace the CRC in the publicly readable registration number with the page number of the page to be scrambled and then encrypt it with a standard block cypher encryption algorithm (such as DES), using a secret encryption key. This produces a 64-bit encrypted number which is unique to each page of each part and is known only to the reader.
2. Divide the 64-bit encrypted number by 256 to obtain a quotient and a remainder that lies in the range 0-255. The value of the remainder gives the position of bit 1 in the scrambled data, and leaves 255 other bit positions unfilled.
3. Divide the quotient from step B by 255 to obtain another quotient and a remainder that lies in the range 0-254. The value of this remainder gives the position of bit 2 in the remaining 255 bits that were unfilled after step B.
4. Repeat step C for each successive bit, decreasing the divisor by 1 each time, until all 64 bits have been placed in their scrambled positions. Each time the quotient reaches 0 during this process, replace it with the original encrypted number from step A.

The steps B-D above are numerically intensive and may not be suitable for microcontroller-based equipment. A simpler but less secure technique is to start with an initial secret, randomly chosen permutation and then further permute it based on the 64-bit encrypted number by interchanging certain bits or not, depending on whether a bit in the encrypted number is a 1 or a 0. This method provides a simpler set of permutations but may still provide adequate security in many applications. The complexity of the technique used to derive permutations from the unique registration number can be selected based on the degree of security needed in the application and the amount of computing power available in the equipment.

V. Description of Operation

Using the methods described above, the automatic debiting equipment operates as follows to decrease the value of the memory by one monetary unit:

1. The equipment detects an Add-Only iButton by means of the presence pulse that it generates, reads the unique registration number, and checks its validity with the CRC.
2. The equipment reads the status registers to find the first page that has not been used up. It then reads that page, making use of the built-in CRC calculation circuitry to confirm the validity of the read.
3. Using a secret encryption key that can be changed periodically, the equipment applies a standard encryption algorithm (such as DES) to the unique registration number (with the CRC replaced with the active page number) to generate a unique secret number, and then uses this number to reorder the bits read from the active page using any of the techniques described in section IV above.
4. After the above reordering, the 0 bits starting from the least significant bit represent credits, and the 0 bits starting from the most significant bit represent debits. The data, beginning with the least significant bit, should therefore appear as an unbroken sequence of 0 bits (credits), followed by an unbroken sequence of 1 bits (not yet used), followed by an unbroken sequence of 0 or more zero bits (debits). The equipment checks the integrity of these three sequences. If there is a break in any of these sequences or if the number of debits exceeds the number of credits, then there is evidence of tampering and the equipment may take appropriate action (such as recording the registration number, or even sounding an alarm or summoning an official).
5. If the number of credits is greater than the number of debits, the equipment adds one more 0 bit to the unscrambled sequence, checks to make sure that the page has not been used up, and then uses the bit permutation in reverse to determine where the debit bit falls in the original scrambled bit sequence. Any time a page is filled, the equipment writes the status bytes to mark the page as used up and proceeds to the next page.
6. The equipment performs a write operation to write the bit identified in step E above from a 1 to a 0, then reads back the page to make sure that the write operation was completed correctly. When a successful write of the debit bit is detected, the equipment activates a peripheral device (passenger gate, etc.) to signal a completed successful operation.

The operation of the crediting equipment is similar to that described above. The crediting equipment receives cash from the user and sets one or more credit bits to 0 to indicate the amount of added value. When a page is half full of credit bits, the equipment proceeds to the next page to add additional credits. The bits are written in the scrambled order so that it is impossible to distinguish the credit bits from the debit bits and the bits that have not yet been used.

Both the debiting and crediting equipment can make use of a secure microprocessor (such as the DS5002 secure micro) so that even if the equipment is stolen, it cannot be made to reveal the secret encryption key which is used in step C above. This makes it possible to limit the knowledge of this information to a very small number of individuals. It is important to note that a blank Add-Only iButton has no monetary value until it his been credited with monetary equivalents using its unique bit scrambling algorithm. Therefore, there is no advantage to a counterfeiter to obtain a supply of blank iButtons, and it is unnecessary to take special precautions to safeguard these supplies.

Assuming that a high-performance processor is used so that the time required to perform the calculations described above can be neglected, the minimum time required for a debiting transaction is the time required to read the unique registration number, read the status bytes, read the appropriate page, and write out the bit that represents the debit. This time, equal to 31.7 milliseconds, is scarcely perceptible and would be regarded as essentially instantaneous by the user.

VI. Summary

Add-Only iButton has the following special characteristics which make it uniquely suitable for applications requiring secure crediting, debiting, and portable storage of monetary equivalent data:

1. A unique, unalterable registration number which allows the data on each different part to be encrypted differently. This makes it impossible to determine how to counterfeit a part by studying how data is written into a different part.
2. Random-access memory which is one-way alterable, that is, having bits that can be changed from a 1 to a 0 but not from a 0 back to a 1. This makes it impossible to write into a part the data pattern it held earlier when it was more valuable. (This type of memory is commonly referred to as onetime-programmable EPROM, but this terminology is misleading in the current application because it suggests that the part can be written only once).
3. A small, durable MicroCan package with a simple electrical connection that allows data to be read or written with a momentary contact.

More Information

DS1982: QuickView -- Full (PDF) Data Sheet -- Free Samples

DS1982U: QuickView -- Full (PDF) Data Sheet

DS1985: QuickView -- Full (PDF) Data Sheet -- Free Samples

DS1985U: QuickView -- Full (PDF) Data Sheet

DS1986U: QuickView -- Full (PDF) Data Sheet