It is a further object of the present invention to provide a system that enables a test taker of a computer administered test on a game computer, where the test is not provided on-line, the test software residing or associated with the game computer, to have his or her test score certified with a central computer and to obtain a ranking and rating with respect to other test-takers.

It is another object of the present invention to provide a system for certifying outcomes of computer generated games played on game computers, and for ranking and rating the players of such games based on their outcomes or an aggregation of their outcomes, with respect to other players of the games, with a central computer having a database storing a unique attribute or identifier for each game computer or software, by generating authenticatable messages on the game computer that represent the players' game outcomes and the unique attribute or identifier associated with the particular game software or the game computer, and authenticating the authenticatable message at the central computer using cryptographic protocols.

It is a further object of the present invention to provide a system for providing cash prizes or other awards or tokens of recognition for players in accordance with their certified ranking and/or rating as described above.

It is another object of the present invention to provide a system for certifying times to completion for races of skill played on game computers which start at designated times, either in connection with a given tournament or independent thereof, where the first participant to complete the game and have his or her time of completion certified by the central computer is declared the winner, and for enabling the participants to be ranked and rated with respect to each other.

It is a further object of the present invention to provide a system for races of skill tournaments, where the start times of the games are variable and players are ranked by the length of time it takes to finish playing the games as determined by a clock associated with the game computer or an external clock signal broadcast over a mass communications means, where the time is authenticated at the central computer and the player finishing a given game in the shortest amount of time is declared the winner.

It is yet another object of the present invention to provide a system for rating/ranking players in tournaments engaged in races of skill as described above, where the players obtain scores for the games where these scores are adjusted by the amount of time it took to complete the games and/or any other play conditions, at the central computer.

It is still another object of the present invention to provide a system for rating/ranking players in tournaments where groups of players form teams and the team scores are certified and ranked at the central computer.

It is a further object of the present invention to provide a system in which players engage in tournaments on game computers, where a start message which enables tournament play contains variables which are read by the game computers and direct the game programs to set game parameters based on player's individual ratings or other parameters, with certain specified attributes or other programmed characteristics, e.g., difficulty, variability, randomness, etc.

It is another object of the present invention to provide a system in which players engage in tournaments on game computers where the players decide when they want to enter the tournaments and play.

It is a further object of the present invention to provide a system in which players engage in tournaments on game computers and where hardware security and/or cryptographic protocols are utilized to ensure the fairness and integrity of the tournament.

It is yet another object of the present invention to provide a tournament system using cryptographic and other protocols, where a trusted third party is not required to prevent undetected player substitution.

It is another object of the present invention to provide a system where the outcomes of computer games of chance are submitted to a central authority and certified using cryptographic and other protocols.

It is still another object of the present invention to provide a system in which players of video games having different ratings/skill levels may play head-to-head matches where the playing conditions during the game are equalized in response to handicap codes.

It is a further object of the present invention to provide a system wherein a computer generated result or outcome obtained on a computer is incorporated into an Authenticatable Outcome Message by the computer, and may be subsequently authenticated on the computer with cryptographic protocols.

It is yet another object of the present invention to provide a system in which a computer generated result or outcome obtained on any computer in the system is incorporated into an Authenticatable Outcome Message by that computer, and may be subsequently authenticated on any other computer in the system with cryptographic protocols.

It is still another object of the present invention to provide a system in which all data in connection with recreating a game played on a game computer may be stored on removable data memory media in an authenticatable format and subsequently used to generate a replay of the game on any game computer in the system by authenticating the data using cryptographic protocols.

It is yet another object of the present invention to provide a system in which a device placed between a game computer and a TV, reads the data in a video output signal to obtain an outcome for the game from the video output signal, and incorporates the outcome into an Authenticatable Outcome Message.

It is still another object of the present invention to provide a system in which a device compatible with a VCR is placed between a game computer and a TV, reads the data in the video output signal, converts the data to digital format, makes the data authenticatable using cryptographic protocols, and stores the authenticatable data in data memory media for subsequent authentication and play back.

It is yet another object of the present invention to provide a pay-per-use system for enabling video arcade type play on home game computers.

It is still another object of the present invention to provide a pay-per-use system for enabling time-dependent disablement with cryptographic protocols of game computers and/or game software.

It is yet another object of the present invention to provide a novel multi-functional game controller for implementing the foregoing with existing game console-type game computers.

In accordance with the foregoing objects, the present invention comprises a system for authenticating the outcomes of computer generated games played on game computers, and for certifying those outcomes as being accurately reported and fairly achieved. The system provides for such certification in connection with tournaments

or independent thereof. The system generally comprises, in one embodiment, a plurality of game computers, where each game computer includes associated memory and a processor for executing programs from its associated memory. The term "associated memory" is intended to include the internal read only memory ROM and read-write memory RAM of the game computer, as well as external devices such as hard disk drives, CD-ROM drives, floppy disk drives, game cartridges and the like. This memory is generally insecure, and may also be referred to as an insecure data source. The game computer contains game software including at least one game program that is executed by the processor to enable a player to play a game on the game computer. The games may be games of skill, races of skill, games of chance, predictions on future events of which the outcome is uncertain, and the like. In a game of skill, the game has an outcome as a result of game play, where the outcome is defined as the entire set of results of the game, including a score, time to completion, all data relating to the game itself, and any play related data. In the present invention, the outcome of the game is incorporated into an Authenticatable Outcome Message AOM that may be subsequently authenticated on the same game computer itself, any other game computer, or by a central computer. In some embodiments described herein, the authentication process not only authenticates but certifies the outcome as being accurately reported and fairly achieved.

An authentication means for generating and authenticating authenticatable messages is operatively associated with the processor of the game computer. The authentication means comprises what is referred to herein as an encryption/decryption module that utilizes cryptographic protocols. The encryption/decryption module may be part of the game software disposed in the associated memory of the game computer, or dedicated firmware disposed within the game computer. Preferably, however, the encryption/decryption module resides within a secure perimeter or security token as described in detail below. The Authenticatable Outcome Message may include data that reveals if the game software has been tampered with by the player. This data is also generated, checked and verified using cryptographic protocols, and is described in more detail below. An authenticated outcome that is determined to have been achieved without cheating the game software or the game computer is certified. The Authenticatable Outcome Message generated by the encryption/decryption module may be subsequently authenticated on the same game computer, on any other game computer with an encryption/decryption module, or by a central authority on a central computer.

The central computer includes an associated memory, a processor for executing programs from the central computer associated memory, and central computer authentication means operatively associated with the processor of said central computer for generating and authenticating authenticatable messages. The central computer authentication means are operable to authenticate Authenticatable Outcome Messages to authenticate game outcomes in response to authentication requests. By checking data appended to the outcome, the central computer can ascertain whether a player obtained the outcome by "cheating" the game software. The central computer may contain a plurality of relational databases for both certifying scores and managing tournaments. The procedures invoked to implement these functions are described in detail below and depicted in the accompanying drawings.

Where a central computer is used to certify outcomes and manage tournaments, communications between the game computers and the central computer may be transmitted via a telephone network. The telephone network may enable communication with live operators, but is preferably coupled to Interactive Voice Response Units IVRUs. The IVRUs are employed to prompt players to enter required information in connection with registering for tournaments and/or for submitting outcomes embodied in Authenticatable Outcome Messages for certification. Alternatively, the game computers may establish an on-line connection to the central computer for the purpose of transmitting registration data and Authenticatable Outcome Messages. The on-line connection may take place over a data network including commercial on-line service providers, Internet, World Wide Web, bulletin board systems or over RF, cable TV, satellite links and the like.

Another aspect of the invention provides for pay-per-use of the game computer or game programs that are executed on the game computer. The pay-per-use system includes a meter that communicates with the game computer, and operates to enable operation of the game computer or execution of game programs upon authorization from the central computer. The meter is a secure device, a computer having hardware disposed within a secure perimeter, capable of generating and authenticating authenticatable messages as described above. In a preferred embodiment, the meter controls operation of the game computer and/or game programs using cryptographic protocols.

In the inventive system, the operating system program of the game computer and game programs, are referred to as metered programs. Each metered program is comprised of a Software Control Block, an Insecure Software Component, and a Secure Software Component. In a first embodiment, the entire metered program resides in an insecure data source associated with the game computer, such as a hard disk or the like. The Secure Software Component is a cryptographically secure set of software instructions, that are decrypted by the meter and executed on the meter to produce at least one output parameter upon which the Insecure Software Component depends, in order to execute the latter on the game computer. The Software Control Block contains information about the metered program that identifies it to the meter, and, in some embodiments, enables the meter to calculate costs for running that program. The meter decrypts and executes the Secure Software Component as long as it has authorization from the central computer, in the form of a time or cost limit.

The many aspects of the present invention will best be understood as the detailed description thereof proceeds with particular reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1**A is an overall schematic of the inventive system in one embodiment;

FIG. **1**B is an overall schematic of the inventive system in a self-authentication and mutual-authentication embodiment;

FIG. **2** is an overall schematic of the inventive system in another embodiment;

FIG. **3** is an overall schematic of the inventive system in still another embodiment;

FIG. **4**A is schematic of the memory arrangement and general components of the game computer;

FIG. **4**B is a schematic of a game cartridge in one embodiment;

FIG. **4**C is a schematic of a secure perimeter for the encryption/decryption module;

FIG. **4**D is a schematic of a game cartridge in another embodiment;

FIG. **4**E is a schematic of a game cartridge in still another embodiment;

FIG. **4**F is a schematic of a game cartridge in yet another embodiment;

FIG. **4**G is a schematic of an embodiment utilizing a secure perimeter and VCR in connection with a game console type game computer;

FIG. **4**H is a schematic of the secure perimeter/VCR interface;

FIG. **5** is a flow-chart of various Authenticatable Outcome Message generation protocols;

FIG. **6**A is a schematic of an exemplary software integrity check;

FIG. **6**B is a flow chart of the software integrity check in the embodiment depicted in FIG. **6**A;

FIG. **7** is a schematic of an exemplary memory arrangement and some hardware for the central computer;

FIG. **8**A is a flow-chart of an exemplary tournament entry procedure;

FIG. **8**B is a schematic of an arcade implementation;

FIG. **9** is a flow-chart of game play;

FIG. **10**A is a flow-chart of an illustrative outcome submission and certification sequence;

FIG. **10**B is a flow-chart of an illustrative biometric verification procedure;

FIG. **11** is a flow-chart of a challenge/response protocol;

FIG. **12** is a flow-chart of a Broadcast Start Message sequence in one exemplary embodiment for races of skill; and

FIG. **13** is a flow-chart of an exemplary tournament sequence for head-to-head games;

FIG. **14** is a schematic of a meter for enabling pay-per-use game play in accordance with the present invention;

FIG. **15** is a schematic of metered software for use in a pay-per-use embodiment;

FIG. **16** is a flow chart of an initialization protocol for the meter;

FIG. **17** is a flow chart of an adding a new program protocol for the meter;

FIG. **18** is a flow chart of an authorization from the central computer protocol for the meter;

FIG. **19** is a flow chart of an updating cost information protocol for the meter;

FIG. **20** is a flow chart of a synchronizing clock protocol for the meter;

FIG. **21** is a flow chart of a starting metered software protocol for the meter;

FIG. **22** is a flow chart of a running metered software protocol for the meter;

FIG. **23** is a flow chart of a reporting usage protocol for the meter;

FIG. **24** is a flow chart of an auditing protocol relating to pay-per-use;

FIG. **25** is a flow chart of an outcome authentication protocol using the meter;

FIG. **26** is a flow chart of another outcome authentication protocol using the meter;

FIG. **27** is a schematic of a descrambling pay-per-use embodiment; and

FIG. **28** is a schematic of a metering device incorporated into a video game controller.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference to the several views of the drawings, there are shown several embodiments of a system in accordance with the present invention generally denoted by the reference numeral **10**.

In the embodiment shown in FIG. **1**A, the system is principally comprised of a central computer **12** associated with a central authority, and a plurality of game computers **14**. The term "game computer" is intended to include personal computers ("PCS"), personal digital assistants, coin-operated arcade video gaming machines, television units coupled to game units (e.g., game consoles such as Nintendo, Sega, etc.) and portable game devices (e.g., GAME BOY, GAME GEAR, NOMAD and the like). For the purpose of description, the game computer depicted in the drawings replicates a standard PC. Each game computer **14** contains software and/or firmware (for convenience, all references herein to programs are to "software") which generates games of the type well known in the art. The practice of playing games on the game computer **14** may be classified as a person against game activity. At the conclusion of a game, the player submits the outcome (e.g., a score and/or time to completion, a combination thereof and any other play-related data) to the central computer **12** as described in detail below. However, many popular games allow for players to play against each other on the same game computer **14**, or by establishing an on-line connection between the players' respective game computers **14**. In this case, the outcome of the players' competition is submitted and certified (e.g., player A beat player B with a score of X to Y). In the following description, each game computer **14** includes game software **15** which resides in memory generally identified by the reference numeral **23**. The memory **23** includes read-write memory RAM, read-only memory ROM, and any non-volatile data source of programs associated with the game computer **14**, such as a game cartridge, hard-disk, CD-ROM, PCMCIA card or special flash ROM chip. The specifics of the game software **15** as relates to the present invention will be described in more detail below. The game computer may also have an associated input control device **17**, such as a joystick (shown) as is well known in the art. The input/output device **17** may comprise multiple joysticks or controls for players to play against each other.

The central computer **12** authenticates and/or certifies outcomes and manages tournaments. It is shown schematically in the drawings as a single unit, but may comprise a plurality of networked computers as required. In order to facilitate tournaments on a large scale, it may be required that the central computer **12** be broken up into regional computers, national computers and even a top level international computer. These computers may be interconnected via data networks such as TYMNET, DATAPAC or the like. The national computers poll the regional computers and the international computer polls the national computer for tournament data. Thus, regional tournaments only utilize the regional computers. If national tournaments are to be held, the national computers obtain the required tournament information from the regional computers. In the case of an international tournament, the international computer polls the national computers. Alternatively, several computers on a single level may be arranged so as to periodically verify each other in accordance with known principles. There are

many ways in which the computers can be arranged, so for the purpose of description herein the central computer **12** is referred to as a single unit. As described in more detail below, in another embodiment of the invention, each game computer **14** is capable of authenticating an outcome from another game computer **14**, with the central computer **12** operating as a Key Distribution Center ("KDC"), i.e., a database of encryption keys used for authenticating messages. The central computer **12** may also operate a "central scoreboard," i.e., a database where all certified scores and statistical information on players and teams are maintained. Statistics for a given player may include information on opponents, the time of play, ratings, rankings and the like. The information may be publicly available, or password protected (i.e., available only to those persons with the proper access password, or to those that have attained a certain rating threshold). The information may be made available via the Internet or major online providers, downloaded to game computers, or by mail or telephone.

In the embodiment shown in FIG. **1**B, the game computers **14** are capable of "self-authentication" and "mutual-authentication." Self-authentication means that a game computer **14** can authenticate an outcome incorporated in an Authenticatable Outcome Message AOM which it generated. For example, if a player claims that a score printed on a piece of paper, or stored in a given memory media, is authentic, and this score is embodied in an Authenticatable Outcome Message AOM, the score may be authenticated by authenticating the authenticatable message AOM on the game computer **14**. Similarly, the authenticatable message may be authenticated on any other game computer **14** in the system (i.e., mutual-authentication). The authentication protocols will be explained in more detail below.

Referring again to the embodiment shown in FIG. **1**A, at least one Interactive Voice Response Unit ("IVRU") **16** is associated with a telephone network and coupled to the central computer **12** through a standard interface for access from a plurality of telephones **18**, to enable players to enroll in tournaments and/or to submit the outcomes of the games to the central computer **12** for certification. In certain implementations, a player may register personal information and/or the game software **15** with the central computer over the telephone **18**. Specifically, IVRUs are responsive to both voice and touch-tone signals communicated from the telephones **18**. In this connection, the game computer **14** may communicate with a Dual Tone Frequency Modulator ("DTFM") to generate messages compatible with the IVRUs **16**. An acoustic coupler **115** may be used to receive messages from the telephone **18** in the same manner. Since the operation of the IVRUs **16** and DTFMs are well known in the art, they need not be described in detail herein. The IVRUs **16** may be associated with an automatic call distributor ACD of the type known in the art to balance the call load. During times of peak calls, calls to any IVRU **16** may be routed to a neighboring IVRU **16**.

In an alternative embodiment shown in FIG. **2**, the game computers **14** may communicate with the central computer **12** via a modem **20**. In this regard, the game computers are not considered to be on-line with the central computer during the game. When a player desires to submit his or her outcome for a particular game or time of completion for a race of skill, the game computer **14** dials up and obtains access to the central computer, and uploads the game outcome information. This is discussed in more detail below. In this connection, it is anticipated that the central computer **12** may be accessed via a website **22** on the Internet **24** or over an on-line data network including commercial on-line service providers, bulletin board systems and the like, as shown schematically in FIG. **3**. The process for establishing an on-line connection to a website on the Internet is well known and need not be described here in detail. It is essentially analogous to establishing a direct on-line link between the game computer **14** and the central computer **12**. In yet other embodiments, the game computers **14** may communicate with the central computer **12** over RF, cable TV, satellite links and the like. For example, in an RF embodiment, communications are simply broadcast in an RF format and transmitted between the game computer **14** and the central computer **12**. The same prompting arrangement as with an IVRU **16** may be employed, with the player entering commands instructing the game computer **14** to send a message to the central computer **12** directly through the key pad or joystick of the game computer **14**. Similarly, messages may be communicated over a cable TV link directly to a television interfacing with a game console.

It is also anticipated that communications between the game computers **14** and the central computer **12** can be implemented with a physical data memory device such as a smart card, diskette and the like. The game computer **14**, for example, might store game-related data onto a diskette which the player would be required to mail to the managing authority for inspection at the central computer **12**. Such a procedure might be required in all instances where the player had won a substantial prize, or where cheating is suspected by tournament officials. Moreover, the game computer **14** may communicate with a printer for printing a copy of an outcome, a game screen containing the outcome and any other relevant data such as game statistics and the like, which may be mailed or faxed to the central authority for subsequent certification of the outcome and such data with the central computer **12**.

Referring now to FIG. **4**A, there is shown a schematic of a portion of an illustrative memory arrangement and some hardware for the game computer **14** in the system of the present invention. For convenience, the internal memory **23** of a personal computer **14** is shown. As described above, the memory **23** includes RAM and ROM, and is coupled to a central processing unit ("CPU") **27** in a conventional manner. The CPU **27** and related hardware are typically referred to as a processor. We use the term "associated memory" to indicate that the game computer memory **23** may also be defined to include a non-volatile insecure data source of programs such as a game cartridge, hard disk, floppy disk, PCMCIA card, special flash ROM chip and the like. Secure memory is disposed within a secure perimeter that will be defined below. The processor loads programs into RAM and executes programs from memory in a conventional manner. In the illustrative embodiment, memory **23** contains a game software package **15** comprised of a game program **26**, an encryption/decryption module **28**, a transmission error check module **30**, a secret software or game computer ID ("SSCID") stored in memory area **32** which uniquely identifies the particular game software **15** and/or game computer **14**, a time/date module **33**, and biometric data in memory area **35**. The game software **15** may comprise a single "program," with the individual elements thereof constituting separate routines. For the purpose of description herein, the term game software **15** can be broadly defined to include a plurality of constituent programs, instructions, routines, files, databases, etc. The game software **15** may also have an associated non-secret software serial number SSN, the purpose of which will become apparent below. The transmission error check module **30** is used to process all incoming messages to the game computer **14** to detect manual input-

ting errors, corruption of transmitted data due to communication problems such as line noise and the like, to enable a resend indication or request to be made. The time/data module **33** time-stamps messages using signals from the clock **36**. The biometric data stored in memory area **35** is used for player verification, which is described in greater detail below. A dedicated game computer **14** may have all of its components including its associated memory **23**, CPU **27** and clock **36** housed in a tamper-resistant and/or tamper-evident enclosure to prevent and reveal, respectively, tampering with any of these components. Tamper-evident enclosures include thermoset wraps which, upon inspection, can reveal any attempt to physically open the structure. Tamper-resistant structures may electronically destroy the memory contents of data should a player try to physically open the structure. A secure perimeter is a defined physical area of hardware which is tamper-resistant and/or tamper-evident, as described in more detail below.

The game program **26** generates games of skill of the type known in the art and commonly played in tournaments such as chess, backgammon, bridge, and the like. Other well-known games of skill (e.g., SONIC AND KNUCKLES, VECTORMAN, DONKEY KONG COUNTRY, MORTAL KOMBAT, STREET FIGHTER, etc.) include those played on dedicated gaming machines such as game consoles, in an arcade or other place where such gaming machines reside. The game program **26** may be configured to enable games to be played in a practice mode, in which the outcomes are not certified or part of a tournament. Such practice games may not have the full functionality of tournament games. A practice golf game, for example, might have less complex wind patterns with wind speed and direction being fixed for a given hole. The tournament version may have winds that frequently change, and which may vary depending on the location of the ball. The game program **26** may also be arranged to include teaching modes for instructing players in a manner consistent with the way they play the game or its result.

The game program **26** may compile a statistical database **31** to store tournament game data that specifically relates to the player's actions during the game. For example, the player of a tournament game may have found X treasures, reached Y levels and eliminated Z enemies. This information may be stored and accessed only by the player who enters the proper code or message into the game computer **14**. This message may be the start message which enables tournament play as discussed below. In a further application, certain aspects of the game, such as, for example, a screen or sequence of events where a player performed a certain move or where a particular opponent was defeated, may be stored and indexed in a database by a certain code to enable the player to call up any one of such screens or sequences at a later time by entering the start message associated with that game (in the case of a tournament) or by some other special command. A menu can be generated upon receipt of the start message or command, enabling the player to select and view the desired screens or particular sequences of events in the game.

The game program **26** may generate races of skill. These include puzzles where the player having the quickest time to completion is declared the winner. A crossword puzzle is a classic race of skill in which players compete to be the first to correctly solve the puzzle. Driving games with lap times also represent races of skill in which the shortest time to the finish line is declared the winner. Referring again to FIG. **4**A, to time races of skill, the game program **26** may use a signal from the computer's clock **36** through the time/date module **33** to time-stamp a particular outcome message or to generate a time message which represents the amount of time the player took to complete a given game. In this connection, the clock **36** may be housed within a tamper-resistant and/or tamper evident seal **38**. Preferably, a real-time clock **36'** is disposed within a secure perimeter **300** as described below. In another embodiment described below, the clock **36'** may reside within a dedicated game cartridge **21**.

In yet another application, the game program **26** may generate games of chance where the outcomes of such games are submitted and certified in accordance with the invention.

The outcome of a game is defined as the entire set of the results of the game, including a score, time to completion in the case of a race of skill, or a combination of both. Alternatively, the outcome may be comprised of all data relating to the game itself (i.e., data stored in memory that enables the entire game to be recreated). In a golf game, for example, such data may include each shot that the player takes, which represents a combination of parameters such as current wind speed, club selected, foot placement, force with which the ball is hit, etc. If these parameters are stored to a disk as the game proceeds, it is possible to subsequently recreate the entire game by replaying the stored parameters. For added security, these values may be stored in encrypted form so that the player cannot alter the game data representing such results after the game is completed.

For typical scored games, execution of the game software **15** by the game computer **14** results in an outcome representing the player's score and, optionally, additional game related information such as the number of levels attained, amount of time spent at each level, number of lives lost, number of enemies eliminated and the like. A game may also have multiple outcomes associated with it. In a game of chess, for example, each move may be considered a separate outcome. Each move can be authenticated by the central computer. In some games, an individual's score may be dependent upon the scores of other players. Authenticating one player's outcome thus requires knowing the outcomes of the other players. For a race of skill, such as a puzzle, execution of the game software **15** by the game computer **14** results in an outcome representing the elapsed time it took the player to complete the game and, optionally, other game related information or subsidiary events such as the completion of certain sub-levels in the game and the like.

An outcome may be transformed or incorporated into an Authenticatable Outcome Message AOM (for clarity, a time of completion is transformed or incorporated into an authenticatable time message ATM) by using a variety of cryptographic protocols including one-way hash functions (also known as compression functions, contraction functions, message digests, fingerprints, cryptographic checksums, data integrity checks (DICs), manipulation detection codes (MDCs), and data authentication codes (DACs)), one-way hash functions with encryption keys (also known as message authentication codes (MACs)), digital signatures, and the like, with the encryption/decryption module **28**. The practice of using cryptographic protocols to ensure the integrity and security of messages is well known in the art and need not be described here in detail. For reference, one of ordinary skill in the art may refer to BRUCE SCHNEIER, APPLIED CRYPTOGRAPHY, PROTOCOLS, ALGORITHMS, AND SOURCE CODE IN C, (2d Ed, John Wiley & Sons, Inc., 1996). The encryption/decryption module **28** contains algorithms and keys for encrypting, decrypting and/or authenticating messages. Examples of well-known cryptographic authentication protocols are as follows:

Encryption

Setup: Central computer **12** and game computer **14** share a secret key.

1. Game computer **14** encrypts outcome message with the shared secret key to form an Authenticatable Outcome Message AOM.
2. Communicate Authenticatable Outcome Message AOM to central computer **12**.
3. Central computer **12** reads and decrypts the Authenticatable Outcome Message AOM with the same key.
4. If the message is intelligible, then the central computer **12** accepts the outcome message as authentic.

*Encryption may be implemented with an algorithm such as DES (U.S. Government standard, specified in FIPS PUB 46). Encryption may utilize any of several algorithms known in the art such as IDEA, Blowfish, RC4, RC2, SAFER, etc. See APPLIED CRYPTOGRAPHY.

Message Authentication Code

Setup: Central computer **12** and game computer **14** share a secret key.

1. Game computer **14** hashes outcome message with a MAC and the shared secret key to form an Authenticatable Outcome Message AOM.
2. Communicate Authenticatable Outcome Message AOM to central computer **12**.
3. Central computer **12** reads the AOM and hashes the message with the shared secret key.
4. If the generated hash matches the received hash, the central computer **12** accepts the outcome message as authentic.

*Any of the MAC algorithms, such as, for example, DES, CBC and the like may be applied in this application.

Encryption with Public Key Cryptography

Setup: Game computer **14** has a public-key/private key pair. The central computer **12** knows the game computer **14**'s public key.

1. Game computer **14** encrypts outcome message with the private key to form an Authenticatable Outcome Message AOM.
2. Communicate Authenticatable Outcome Message AOM to central computer **12**.
3. Central computer **12** decrypts the AOM with the public key of the game computer **14**.
4. If the message is intelligible, the central computer **12** accepts the outcome message as authentic.

A sample algorithm for this procedure is RSA.

Signing with Public Key Cryptography

Setup: Game computer **14** has a public-key/private key pair. The central computer **12** knows the game computer **14**'s public key.

1. Game computer **14** signs the outcome message with the private key to form an Authenticatable Outcome Message AOM.
2. Communicate Authenticatable Outcome Message AOM to central computer **12**.
3. Central computer **12** verifies the signature using the outcome message and the public key. The mathematics of verification indicates whether the outcome message is authentic.
4. If the outcome message is intelligible, then the central computer **12** accepts the outcome message as authentic.

There are several ways to ensure that an Authenticatable Outcome Message AOM is "fresh" (i.e., it has been used more than once). In the first, known as "challenge/response", the central computer **12** generates a random or sequence number (also referred to as a "nonce") and communicates it to the game computer **14**. The game computer **14** then incorporates this random number in the Authenticatable Outcome Message AOM. If the random number received matches the random number just generated, the central computer **12** accepts the message as fresh, i.e., an old message would contain a different random number.

In another method, the game computer **14** includes a sequence number in the Authenticatable Outcome Message AOM. This sequence number is incremented by one every time the game computer **14** generates an Authenticatable Outcome Message AOM. The central computer **12** stores the most recent sequence number in memory. It accepts the current outcome message if the sequence number received is one greater than the stored sequence number.

In yet another implementation, the game computer **14** includes the current time in the Authenticatable Outcome Message AOM. The central computer **12** then checks the time associated with the Authenticatable Outcome Message AOM against the time from the central computer's associated clock. If the times are within a prescribed window, the current outcome message is accepted as fresh.

In still another procedure, the game computer **14** includes a random number in the Authenticatable Outcome Message AOM. The central computer **12** maintains a database of all random numbers received from the game computers **14**. If the new random number is not in that database, then the current Authenticatable Outcome Message AOM is accepted as fresh. If a time element is incorporated as well, then the central computer **12** only has to store a relatively small quantity of unexpired messages.

In FIGS. **4**A and **4**B, the encryption/decryption module **28** is depicted as part of the game software **15**. In that embodiment, the encryption/decryption module **28** refers to cryptographic algorithms and keys or other data, software instructions and the like. These reside with the rest of the game program in a data storage device such as a game cartridge, hard disk, CD-ROM or the like. The actual processing to implement the cryptographic protocols takes place in the game computer's CPU **27**, not within a specialized cryptographic processor.

Preferably, as shown schematically in FIG. **4**C, some or all of the encryption/decryption module **28** resides within a secure perimeter **300**. A secure perimeter is a defined physical area of hardware which is tamper-resistant and/or temper-evident, in which resides data or algorithms whose characteristics must not be alterable in order for a system to remain secure. Examples of devices which incorporate secure perimeters include U.S. military encryption devices such as the STU-III telephone made by Motorola and AT&T, and the iPower card, available from National Semiconductor Corp. The latter is a dedicated encryption/decryption device embodied in a PCMCIA card which can interface with the game computer **14** through, for example, the game computer's PCMCIA socket (for game computers **14** with PCMCIA compatibility). In the iPower card, both the key/data storage and cryptographic functionality are located within the secure perimeter. Referring again to FIG. **4**C, a secure perimeter such as an iPower card is shown. It includes a 32-bit CPU **302** with ROM **304** containing encryption algorithms, a real-time clock **36**' and an interface with an off-chip battery **(310)**—backed RAM **308** which holds encryption data and key information. Any attempt to tamper with or get at the encryption data stored within the device results in a memory loss of that data. Moreover, the I/O pins **312** of the device are electrically isolated to prevent pin-level probes, and the chip

itself contains mechanical and chemical protection to prevent chip-probing equipment from accessing the encryption information from the CPU **302** directly. Such a secure perimeter **300** may additionally include additional non-volatile memory **313** for storing other software instructions and/or additional game related or other data as will be explained in more detail below. Hereinafter, the CPU **302** within the secure perimeter **300** will be referred to as the secure CPU **302**, and the memory within the secure perimeter **300** will be referred to as the secure memory. Thus, within an iPower card, all encryption/decryption functions are performed in the secure perimeter and do not take place in the CPU **27** of the game computer **14**. Communications between the secure CPU **302** of the secure perimeter **300** and the CPU **27** of the game computer **14** are known in the art and need not be described here in detail. When the secure CPU **302** of the secure perimeter **300** is called upon by the game computer **14** to generate an Authenticatable Outcome Message, authenticate an Authenticatable Outcome Message, and/or perform any other required functions, the CPU **27** of the game computer **14** sends the appropriate signals to the secure CPU **302** of the secure perimeter **300**. In this regard, the entire encryption/decryption module **28** is said to reside within the secure perimeter **300**. This means that all cryptographic keys and algorithms are stored in memory within the secure perimeter **300**, and all cryptographic functions are implemented on the secure processor **302** within the secure perimeter **300**. Thus, when cryptographic protocols are to be used for encryption and/or authentication, the game computer CPU **27** communicates commands and data to be encrypted or made authenticatable to the secure processor of the secure perimeter **300** as described below, requesting that the data be cryptographically processed. The secure CPU **302** of the secure perimeter **300** may be used to subsequently authenticate the Authenticatable Outcome Messages AOMs that it generates, as well as Authenticatable Outcome Messages AOMs from any other game computer **14** in the system. It may also be used to time-stamp messages or track times to completion for races of skill with the clock **36**'. The secure CPU **302** may also perform some of the computational tasks required to execute the game software.

In an alternative embodiment, cryptographic keys may be stored in secure memory within the secure perimeter **300**, but cryptographic algorithms and software instructions are stored in unsecure memory associated with the rest of the game software **15** (i.e, in a conventional game cartridge, hard disk, floppy disk, CD-ROM or the like), and actual processing to implement the cryptographic protocols takes place in the game computer's CPU **27**.

External secure devices such as the aforementioned iPower cards can function as "tokens." A token is a physical computing device used by individuals to gain access to protected electronic resources. Tokens commonly include cryptographic capabilities and can store keys or other data. Intelligent security tokens may be utilized to prevent unauthorized player access to the game computer **14**, as well as for implementing the encryption/decryption functions for outcome authentication and certification. The iPower card described above, is an example of a token contained within a secure perimeter.

Other such tokens include the SMARTDISK, manufactured by SmartDisk Security Corporation. The SMARTDISK contains a CPU and memory used for encrypting and decrypting data. Thus, as with the iPower card, the encryption/decryption module **28** may reside in the SMARTDISK (i.e., all cryptographic functions are implemented within the SMARTDISK). The SMARTDISK requires a user password to function. Thus, access to the system requires the player to physically possess the token and know the proper password. Smart cards are similar tokens. They are shaped like credit-cards, but contain an embedded microprocessor for implementing various security functions.

Another type of token called TOUCH MEMORY is manufactured by Dallas Semiconductor Corporation. This device consists of a computer chip housed within a small button shaped stainless steel case. The case may be ring-shaped and worn around a player's finger. The chip contains up to 64 kb of RAM or EPROM, sufficient to store a plurality of cryptographic keys. The device transmits data bidirectionally at 16.3 kb per second when placed into contact with a reader device. Each chip contains a unique serial number that is laser-etched into the chip at the time of manufacture. Keys from the device may be used in any of the cryptographic protocols described herein for authentication and/or encryption, as well as for user identification. The DS1422 UNIQUEWARE product can be configured to transparently decrement each time that the device is used, allowing players to obtain and store a limited number of start messages, for example. The DS1427 configuration includes a tamper-resistant real-time clock **36**' that may be utilized in the different applications described herein.

In yet another embodiment, a player may obtain a joystick which has a unique identifier associated with it. The joystick identifier may be used as a key in the cryptographic protocols described herein, or to enable the player's game software **15** to generate a certified or tournament game. The key may be stored on a ROM chip within the joystick. When the game software **15** loads instructions to generate the game into the RAM of the game computer **14**, the key from the joystick is loaded into RAM and verified. If the proper key is not found, the game software **15** may be disabled. This is conceptually similar to a "dongle." Alternatively, the joystick may have an associated input/output interface for accepting data from and communicating data to a secure perimeter such as an iPower card. Thus, the authentication protocols may take place within the structure of the joystick. This approach is described in more detail in an alternative embodiment below.

Other secure devices include the SENTINEL SUPERPRO manufactured by Rainbow Technologies. The SENTINEL SUPERPRO is a dongle which stores keys required for the operation of software applications. The software directs the computer it is running on to access the dongle and, if it does not find the right key, suspends execution of the program. The dongle plugs into the parallel port or ADB port of a personal computer and measures 1.65 inches long by 2.125 inches wide. It contains 128 bytes of read/write memory organized as sixteen 64 bit words. These words can be used as time counters for leased software, or they can count executions for limiting the operation of demonstration products. Memory cells can also be programmed with customer information, serial numbers, passwords, and the like. With regard to the present invention, the keys and algorithms that form part of the encryption/decryption module **28** could reside in the memory of such a dongle.

Dongles may also include specialized cryptographic processors and memory, allowing functionality similar to that found within an iPower card. Dongles configured to plug into a parallel port, however, have the advantage of being compatible with nearly all Intel-based computer hardware, as opposed to iPower cards which require PCMCIA capabilities.

The above secure devices are particularly well suited to the storage of game related data for auditing purposes. In a

computer golf hole-in-one tournament, for example, it may be desirable to track each swing that a player takes since large prizes for a hole-in-one would attract hackers interested in forging such an event. To prevent such cheating, game parameters (swing speed, club used, etc.) would be sent to the secure CPU **302** where they would be encrypted. This encrypted data could be stored in the non-volatile secure memory **313** within the secure perimeter **300** as an encrypted receipt file. Any player scoring a hole-in-one could be required to send in the secure device before receiving payment, allowing tournament officials to examine the game data to see if it matched the claimed result. Alternatively, the encrypted game parameters could be communicated back to the central computer **14** and stored on the hard drive or copied to a floppy disk(insecure memory). In the event of a claim for a large prize, the player would simply mail in the disk to the managing authority and the encrypted data would be decrypted and analyzed by the central computer **12** by recreating the game with such data to determine whether the claimed score was actually achieved.

For an additional level of security, the secure CPU **302** may perform some of the game calculations normally executed by CPU **27**. In an illustrative application, the game program renders a golfing game of skill, such as, for example, PGA TOUR 96 available from ELECTRONIC ARTS. In this game, a digital image of a golf game is rendered on the game computer **14**, comprising a golf ball on a tee, fairway, trees, sandtraps, etc. A human figure is superimposed on this background, and swings a golf club in response to player inputs via a keyboard or joystick. The player's club swing data represents various parameters, including the club selected (e.g., one iron, two iron, three wood, etc.) and its specific characteristics (e.g., club head orientation), foot placement, and swing force, speed, direction and the like. In the course of a typical computer generated golf game, these parameters are applied to software instructions that compute a trajectory path for the ball to generate a resultant ball location. After the player swings the club, the display may depict the new ball location relative to the hole. Other factors, including ambient conditions such as wind speed and direction or other random variables, may be introduced for greater realism. With a secure CPU **302**, the calculation as to the new location for the ball may be taken away from CPU **27**. This is accomplished by making these calculations part of a Secure Software Component **710** of the game program **26**, which will only run on the secure CPU **302**, for example, by encrypting the block of software instructions that relate to computation of such portions of the game with the above-described game parameters, or by requiring additional data or algorithms stored in secure memory within the secure perimeter **300** to make such computation. Thus, the secure CPU **302** computes the new ball position, which is then communicated back to the CPU **27** of the game computer **14** where it is displayed on screen. The results of the calculations can also be stored on the hard drive of the game computer **14** as described previously. Some game variables, such as wind speed, could also be generated by the secure CPU **302**. These variables would impact the calculation of the new ball position, and would prevent players from using mechanical devices to play a perfect game since at least one variable cannot be controlled. Alternatively, a game parameter such as wind speed, could be generated by the secure CPU **302**, and then transmitted to the CPU **27** where the new ball position could then calculated. This embodiment is described in more detail below with regard to the pay-per-use metering system.

For game console applications, a secure CPU **302** within a secure perimeter **300'** may be adapted to interface with a VCR unit **400** as shown schematically in FIGS. **4**G and **4**H, to enable cryptographically protected recording and playback of games generated on the game computer **14**. In this embodiment, the video output signal from the game computer **14** is communicated to a video input on the VCR **400**, and a video output signal from the VCR **400** is communicated to a television **402** in a conventional manner. The secure CPU **302** and associated hardware within secure perimeter **300'** are configured to fit into a standard VCR slot **404**. In addition to the secure CPU **302**, the secure perimeter **300'** includes ROM **304** containing encryption algorithms, a real-time clock **306** and an interface with an off-chip battery **(310)**—backed RAM **308** which holds encryption data and key information. The secure perimeter **300'** further includes interface circuitry **406** for communicating signals from the read/write head **408** of the VCR **400** via an analog/digital ("A/D") converter **410** to the secure CPU **302**. Video information is typically communicated to the television **402** in an RF format. The RF video signal may be processed in the VCR **400** by the front-end receiving circuitry **412**, which demodulates the video signal to a base-band signal as is well-known in the art. Normally, the demodulated information is what is recorded on a VCR tape cassette. In the inventive application, the base-band video signal data is converted to digital format by the A/D converter **410**, encrypted with a private key, and stored in the non-volatile memory such as an EPROM **414**. For playback, the secure CPU **302** authenticates the game data, for example by decrypting the data with the corresponding public key, and the authenticated game data is then processed to generate a video signal. The secure perimeter **300'** may also contain software instructions in ROM for generating an Authenticatable Outcome Message AOM to be used as described hereinbelow. This Authenticatable Outcome Message AOM may be included in the video signal to appear on the television screen at the end of the game.

Referring now to FIG. **4**B, there is shown a schematic of a game cartridge **21** for use with the system of the present invention. The game cartridge **21** includes a housing **19** that contains the game software **15** in a ROM **23***a* built into the cartridge, and the ROM **23***a* interfaces with the game computer **14** via an I/O interface **25** in a conventional manner. The software serial number SSN may be displayed on the exterior of the cartridge housing **19** as shown. The game software **15** in the case of typical games such as those offered by Sega and Nintendo, includes a game program **26** which offers the player a choice of a tournament enabled game or a non-tournament enabled (regular) game. Tournament enabled games may be generated with the "cheat codes," typically used by developers in testing the game, disabled in the game program **26**. In addition, certain play aspects of the game which usually occur in some known sequence or have some known characteristics (such as the location of bonuses or certain challenges), may be changed in the tournament version of the game to ensure that the game is less predictable than that of the regular version. While the regular version of a computer golf game may have only a few sand traps, the tournament version may have many. Opposing pitchers may throw the ball at 80 miles per hour in a regular computer baseball game, while in the tournament version opposing pitchers throw at 100 miles per hour. Game cartridges may also contain game software **15** configured for one-time or limited time use.

Referring now to FIG. **4**D, a game cartridge **21** contains the game software **15** in volatile memory **23***b*. The volatile

memory **23***b* is connected to the I/O interface **25** in a conventional fashion. The volatile memory **23***b* is also connected to a power source **27** via a tamper switch **29**. The tamper switch **29** is coupled to the cartridge housing **19**, at the interface shown schematically at **31**, so that any attempt to break open the cartridge housing **19** causes an interruption in power from power source **27** to volatile memory **23***b*, thereby causing all program data stored in volatile memory **23***b* to be erased. The tamper switch **29** may take many forms, depending upon the configuration of the game cartridge **21**. In an exemplary embodiment, the tamper switch **29** is adapted to the cartridge housing **19** such that a physical incursion simply causes the tamper switch **29** to open. Alternatively, the tamper switch **29** may consist of a photocell sensitive to a certain level of light that causes a power interruption if the cartridge housing **19** is opened. In either case, an interruption of power to the volatile memory **23***b* causes all stored program data to be erased. This procedure is well-known in the art for securing computer memory devices. The clock **36** may also be housed within the game cartridge **21** such that any attempt to alter the clock **36** results in a loss of program data stored in volatile memory **23***b*.

Referring now to FIG. **4**E, all game software data (excluding the encryption/decryption module **28**) is encrypted and stored in non-volatile memory **23***c*, while the encryption/decryption keys and algorithms (encryption/decryption module **28**) are stored in volatile memory **23***d*. Thus, any action which triggers the tamper switch **29** causes an interruption in power and the encryption/decryption module **28** stored in the volatile memory **23***d* to be erased. Without the encryption/decryption module **28**, the encrypted data stored in the non-volatile memory **23***c* is useless.

In another embodiment shown in FIG. **4**F, the game software **15** resides in an electrically erasable and programmable read only memory (EEPROM) **23***e*. If the cartridge housing **19** is opened, the tamper switch **29** closes and an erase signal from power source **27** causes the data stored in the EEPROM **23***e* to be erased. The practice of erasing data in an EEPROM is well known and need not be discussed in detail here.

It will also be appreciated that special enhanced security tournament cartridges **21** may be supplied to players for advanced rounds of competition in connection with any tournament.

Referring again to FIG. **4**A, as a means of obtaining information as to where games are being played for compiling various tournament statistics and/or for preventing game play when the game computer **14** resides in certain locations, the game computer **14** may communicate with or have an integral Global Positioning System ("GPS") **37**. A GPS receiver derives positional information from a plurality of satellites. The GPS information may be used to prevent game play in certain locations by providing a location lockout feature in the game software **15**. When the player attempts to begin a game on the game computer **14**, the game software **15** queries the GPS **37** and checks whether the current location of the game computer **14** is within an allowed area. This allowed area may be incorporated into the game software **15**. If the game computer **14** is found to be outside of an allowed area, the game software **15** directs the game computer to deny player access to the game. In a different application, the positional information may be incorporated into the Authenticatable Outcome Message AOM and uploaded to the central computer **12** when a player submits his or her game outcome. In this regard, the central computer **12** can use the positional information for ranking/rating players without requiring submission of the player's specific location (i.e., the home address), and/or for compiling statistical location data. The central computer **12** can ascertain which state, municipality or even town where the game computer **14** was located or, if the player was mobile, all areas where the player was located when the player played the game, either by uploading the information from the game computer or by accessing a database containing such information. Most GPS receivers have the capability to store a sizable amount of data. Typical handheld GPS receivers used in aviation applications can store enough information to save positional data for an entire flight. Although current GPS satellites are subject to having their GPS signals degraded by the military without notice, future civilian systems that are currently under development will be capable of providing consistently accurate positional information to within a few feet.

To preclude player substitution, biometric identification devices such as a fingerprint reader, voice recognition system, retinal scanner and the like, may be used to provide absolute player identity verification at the game computer **14**. An example of such a device is the FC100 FINGERPRINT VERIFIER **31** available from Startek, a Taiwanese company. The FC**100** is readily adaptable to any PC via an interface card **39**. The fingerprint verifier **31** utilizes an optical scanning lens. The player places his or her finger on the lens, and the resulting image is scanned, digitized, and the data compressed and stored in memory location **35**. Typically, a 256 byte file is all that is required. Each live-scan fingerprint is compared against the previously enrolled/stored template. If the prints do not match, access to the system can be denied. This procedure may be implemented before the initiation of a tournament game, during the game in response to prompts from the game software **15** at some predetermined or random times, or continuously by incorporating the scanning lens into a joystick on the game computer **14** such that the player is required to maintain his or her finger on the lens at all times during the game for continuous verification. The fingerprint data may also be registered and stored in the central computer **12** (either in its compressed form or as hash value) in a player information database for player identity verification during various protocols, and/or used as a key as described below.

A voice verification system which utilizes a person's "voice-print" may also be used to provide player identity verification at either or both the central computer **12** and the game computer **14**. The process of obtaining a voice-print and subsequently using it to verify a person's identity is well-known in the art, and therefore need not be described in detail herein. One of ordinary skill in the art may refer to SpeakEZ, Inc. for voice identification/verification technology. Specifically, speaker identification software is utilized to take a sample of the player's voice. This sample is stored in the central computer **12** in the player information database. Each time the player calls the central computer **12**, it prompts the player to speak his or her name into the telephone **18**. The speaker identification software then directs the central computer **12** to check the player's current voice-print against the voice-print stored in memory. If there is a match, the procedure continues. This is described in more detail below. The voice-print may also be stored in a database in the game computer **14**, to verify the player's identity at that location prior to allowing game play without the central computer **12**. This is also described in more detail below.

Referring now to FIG. **5**, there are shown several exemplary ways in which the game computer **14** can generate an

Authenticatable Outcome Message AOM. At the conclusion of the game, an outcome (e.g., a score) is displayed. In one embodiment, the outcome may simply be embodied in a code generated using any secret algorithm. This algorithm is not readily ascertainable or known by the player. It resides in the game software **15** or in a separate encryption/decryption module **28**, and in the central computer **12**. Accordingly, when the player seeks to register an outcome of, for example, 1,000,000 points for game XYZ, the game computer **14** generates an Authenticatable Outcome Message AOM, for example, 21328585, with the secret algorithm. The central computer **12**, the same game computer **14**, or any other game computer **14** applies an inverse of the secret algorithm to the Authenticatable Outcome Message 21328585, or the same algorithm to the score of 1,000,000 points for that game, and if the results match, the authenticity of the outcome is verified. Thus, an outcome cannot be created or guessed without actually playing a game on a game computer **14** containing the secret algorithm. In a preferred embodiment, the encryption/decryption module **28** generates an Authenticatable Outcome Message incorporating the outcome (and any play-related data) using the SSCID as an encryption key. This encryption of the outcome (and play related data) with the SSCID enables authentication of the outcome with respect to the particular game software **15** or game computer **14**. Alternatively, the SSCID is combined with the outcome and the combination is incorporated into an authenticatable message with a different key. In this regard, the encryption of the outcome and SSCID may utilize the biometric data scanned with the fingerprint verifier **31** or obtained from the voice print system as described above as a key. In this manner, the player's identity may be verified in the authentication process. While the secret game software or computer ID, SSCID, is not made known to the player, it is possible to generate a known serial number based upon the secret number. In this connection, after the player powers up the game computer **14** for the first time, it implements a registration process. The SSCID is encrypted by the encryption/decryption module **28** and displayed on screen. The player calls the central computer **12**, and enters the SSCID, along with his or her name and/or PIN. The central computer **12** then decrypts the SSCID and associates it with the player's registration information. The central computer **12** then generates a unique random number RS which is tied to the SSCID. The player writes this number down, and can use the same to identify his or her game computer **14** when authentication is not required. The same procedure can also be used to generate known serial numbers for secret software numbers. Software can also be tied to hardware. A player can be forced to register his new software before he plays the first game. In this regard, the game computer **14** displays the SSCID in encrypted form. The player calls the central computer **12** prior to initiating play. The SSCID is added to the player information database **48**, and is then used in the authentication process of any outcome as described herein. This ensures that the player can only submit an outcome for authentication/certification when using his or her game computer and/or game software **15**. Use of another player's game software **15** and/or game computer **14** will cause the authentication process to fail.

In yet another implementation, an outcome may be represented by other data or symbols which are intelligible only to the central computer **12**, but not to the player. For example, the score 5000 is represented by symbol data comprised of three green dots, four brown squares and two purple triangles. After communicating the score to the central computer **12**, the player is required to send this data for confirmation of the outcome. The player is unable to determine whether this combination corresponds to the same score, a higher score or a lower score. But the central computer **12** is able to decipher these symbols to determine if, in fact, they represent the same outcome submitted for certification in accordance with some secret coding protocol. Alternatively, the player is not provided with an actual score. The score is secret, and is revealed to the player by the central computer only after it interprets the symbol data. This is similar to encrypting or encoding the outcome.

In the case of tournaments, the Authenticatable Outcome Message AOM may prove tournament validity, by including data representing that the outcome was the result of a valid tournament game. This data may constitute a subliminal message within the Authenticatable Outcome Message AOM. Alternatively, the Authenticatable Outcome Message AOM may include all or part of the Authenticatable Start Message ASTM for initiating tournament play for this purpose.

Authenticatable Outcome Messages AOMs may also contain statistical data for enabling the sanctioning authority to compile market research information. This data may be compressed by the game computer **14**, and decompressed by the central computer **12**.

The game software **15** may be adapted to instruct the game computer **14** to save game play up to a certain point in a game, and to resume play from that point at a subsequent time. In this regard, a "resume code" may be generated, which enables a player to pick up a game from where he left off. The game play to a specific point may be stored entirely in non-volatile memory. This would allow golf tournaments in which players could stop after a number of holes had been completed, picking up play at a later time or date. Alternatively, the game computer **14** may generate an Authenticatable Outcome Message AOM that represents the game outcome to this point. This allows for a first player to "hand off" the AOM to the next player who continues the game. Such an arrangement is analogous to a relay-race scenario where a player runs a certain distance and then hands off an object to the next runner. It also enables the same player to resume game play without having to store the large amount of data representing the game play to the point of termination. Since game programs generate games that are typically segregated into various levels, where the player advances from level to level as the game proceeds, this "code" may be used to instruct the game software **15** to continue from any given point. When the player selects a "quit" or "end game" option, if the player desires to continue the same game at a subsequent time, he inputs the Authenticatable Outcome Message AOM into the game computer **14**. If it is authenticated, then game play proceeds from the prior termination point.

To prove integrity of the game software **15** through the outcome certification process (i.e., that it has not been tampered with), digital signature protocols may be utilized. In this regard, a digital signature algorithm with a private key is employed to "sign" a message. This message may be a hash value of the software generated with a function, a compressed value of the software code produced by a compression algorithm, and the like. The signed message is then verified using the digital signature algorithm with a public key at the central computer **12**, the same game computer **14** or any other game computer **14** in the system. The secret key may reside in the encryption/decryption module **28**, and preferably, the encryption/decryption module **28** resides in a secure perimeter **300** as discussed above. The secret key may be the SSCID, and/or a hash or com-

pressed value of the digitized biometric fingerprint data or voice print described above. The public keys may be contained in the KDC at the central computer **12** as mentioned above, to enable players to verify the digital signature of the software at their respective game computers **14**.

In an exemplary embodiment, the encryption/decryption module **28** generates a hash value of the software instructions which make up all or part of (i.e., game program **26**) the game software **15**. This hash value is incorporated into the Authenticatable Outcome Message AOM. The hash value is generated using a one-way hash function which operates on a numerical representation of the game software **15**. An example of a one-way hash function is the Secure Hash Algorithm ("SHA"). SHA is a U.S. government standard, and is specified in FIPS PUB 180 of the National Institute of Standards and Technology. Other examples of hash algorithms include MD4, MD5, RIPE-MD, Haval, etc. One skilled in the art may refer to APPLIED CRYPTOGRAPHY. As a specific example, each character of the game software **15** may be converted to ASCII values and then into a binary series of 1s and 0s. An exemplary one-way hash function may operate on this series as follows: (1) exchange the positions of all 1s and 0s; (2) group the digits into blocks of 64 digits each; (3) raise each block to the 5th power and then truncate the result to 64 digits; (4) take the final complete number and square it; (5) convert this binary number to base ten; and (6) take the last 24 digits as the hash value. The initial hash value for any given copy of the game software **15** is created prior to sale or distribution, and may be stored in the central computer **12**, or even publicly known. This hash value may be derived from a different one-way hash function for each copy of game software **15** sold. If the player attempts to alter the game software **15** by tampering with the software instructions to produce a more favorable game outcome (i.e., a higher score or faster time to completion), such modifications to the game software **15** will be evidenced by the mismatch between the newly generated hash value and the initial hash value stored in the central computer **12**.

Since the game software **15** may be tampered with while the software instructions reside in the volatile read-write memory (i.e., RAM) of the game computer **14**, tampering may not be detected by just generating a hash value of the entire game software **15** at the end of a game. One way to detect and provide evidence of tampering of the game software **15** while it resides in RAM, is to have the secure CPU **302** of the secure perimeter **300** periodically check blocks of the game software **15**. In this connection, referring now to FIGS. **6**A and **6**B, the game software **15** may be configured with n blocks **314** of instructions, where each block **314** has an associated hash value $h_1 \ldots h_n$ determined by using a one-way hash function. Similarly, a master hash value h_m of all the block hash values $h_1 \ldots h_n$ is also determined using a one-way hash function. These values may be stored in the secure memory (ROM **304** or other non-volatile memory **313**) of the secure perimeter **300**. The secure memory of the secure perimeter **300** may store such values for many different games. At step **316**, a block is loaded into the RAM of the game computer **14**, and its instructions are executed at step **318**. When the block **314** of software instructions is to be replaced in the RAM of the game computer **14**, that block **314** is read by and loaded into the secure RAM **308** of the secure perimeter **300** at step **320**. The secure CPU **302** calculates a hash value hsp_n of that block **314** using the one-way hash function at step **322**, and the computed hash value hsp_n is compared to the known hash value h_n for that block **314** stored in the secure memory of the secure perimeter **300** at step **324**. If the computed block hash value hsp_n matches the expected value h_n, and the game is not over at step **328**, the next block of instructions **314** that replaces the previous block, represented by incrementing n at step **330**, is loaded into the RAM of the game computer **14**. If a block hash value hsp_n does not match h_n at step **326**, the secure CPU **302** can do several things. It can send a message to the game computer **14** to disable the game program **26** at step **328**. Alternatively, the secure CPU **302** generate a tamper indication which is included at the end of the game in the Authenticatable Outcome Message AOM at step **332**, or, if no tampering is detected, it can generate a non-tampering indication which is included in the Authenticatable Outcome Message AOM at step **329**. Thus, when the player attempts to submit an outcome for certification which was obtained with tampered game software **15** as evidenced by the tamper indication, the central computer **12** can reject the outcome. The secure CPU **302** in the secure perimeter **300** may alternatively calculate a master hash value hsp_m based upon all of the individual block hash values $h_1 \ldots hsp_n$ that were calculated as each block **314** was examined. This master hash value hsp_m may then be compared to the expected master hash value h_m stored in the ROM of the secure perimeter **300**. If the master hash values do not match, the secure CPU **302** in the secure perimeter **300** can generate a tampering indication which is incorporated into the Authenticatable Outcome Message AOM. Alternatively, the master hash value hspm itself may be incorporated into the Authenticatable Outcome Message AOM, and subsequently verified at the central computer **12** as described above.

In the case of a dedicated game computer **14** (i.e., a game console), where it is more difficult to access and alter software instructions while loaded in RAM, or where the requisite level of security is not great, the secure perimeter **300** may not be required. However, there exists a problem unique to game consoles in the form of the GAME GENIE video game enhancing device. The GAME GENIE is an interposing device that connects between a game cartridge **21** and the game computer **14**. In a game console application, the game software **15** resides in the ROM of a dedicated game cartridge **21**. The interposing device enables a player to temporarily change certain game play-features by altering program instructions that are loaded from the ROM of the game cartridge **21** into the RAM of the game computer **14**. These changes are not permanent, and disappear when the power to the game computer **14** is turned off. This provides a unique challenge in the context of the present invention, where the certification aspects rely, in part, on verifying the integrity of the game software **15**. The present invention overcomes the interposing device problem by utilizing one-way hash functions and encryption in authentication protocols.

As in the above example incorporating one-way hash functions, the game software contains n blocks **314** of software instructions, where each block **314** has an associated hash value $h_1 \ldots h_n$ and the entire set of instructions has a master hash value h_m computed by applying the individual hash values $h_1 \ldots h_n$ to a one-way hash function. The hash value of each block may be determined with the same or a different one-way hash function. One of the blocks **314** may contain a list of all hash values for the other blocks, the master hash value h_m, and the hash function or functions used for calculating each block hash value and the master hash value. As described above, the master hash value h_m may be stored in the game software instructions, or can be input by the player into the game computer **14** at the start of

a game. Thus, the master hash value h_m is checked at game start by initially computing the master hash value $hnew_m$ from the values $h_1 \ldots h_n$ to determine whether it matches the value which is either stored in the instructions or input by the player. If the interposing device was used to modify any of the instructions, the computed master hash value hnewm will not match the input master hash value at the game start. To ensure that the interposing device does not subsequently alter software instructions as they are loaded into RAM, the game software **15** contains instructions that direct the CPU **27** to compute the hash value $hnew_j$ of each block j as it is to be replaced in RAM, and a recalculated master hash value $hnew_m$ based upon the new hash value for block j and the known hash values of all the other blocks. The calculated master hash value $hnew_m$ is then compared to the known master hash value h_m in the hash block. If at any time a discrepancy is found, the game software **15** may instruct the game computer **14** to disable the game software **15**, or generate a tampering indication that is included in the Authenticatable Outcome Message AOM. In this connection, the hash values $h_1 \ldots h_n$ and hm, the one-way hash functions, and the instructions for checking the game software instructions in this manner may reside in a ROM chip internally associated with the game computer **14**. Thus, although the protocol is essentially the same, no "security" instructions are executed from the game software **15** itself.

Another solution to the interposing device problem resides in the use of an authentication protocol to enable the game software to run in the game computer **14**. In this connection, the game software instructions stored in the ROM of the game cartridge may be made authenticatable where the game computer **14** authenticates the instructions prior to executing the program. This can be implemented by encrypting the software instructions with a private key by the game developer, thereby requiring that the game computer **14** decrypt the encrypted game software instructions with the corresponding public key prior to execution of the game program. The instructions and algorithm(s) for performing the decryption process reside in a ROM chip (not shown) in the game computer **14**. The game software instructions are encrypted in blocks. Before each block is executed in the RAM of the game computer **14**, it is decrypted by the CPU **27** with the algorithm(s) and keys stored in the ROM chip. If an interposing device is used to make changes to the game software instructions, the authentication process implemented by decrypting the encrypted game software instructions with the public key will reveal unintelligible commands and the game program will be altered. This alteration may be detected by a security program in the ROM chip and used to disable the game software **15** and/or incorporated into the Authenticatable Outcome Message AOM to indicate tampering.

In addition to cryptographic techniques for defeating GAME GENIE-type devices, there are other methods that can be equally effective. One technique is to authenticate not only the score of the game but several key characteristics of the game. A GAME GENIE, for example, might allow a player to be completely invulnerable to the attacks of opponents. If the game he or she were playing required, for example, defeating a dragon at the end, the GAME GENIE enhanced player would have no trouble quickly defeating the dragon. Most players would take a longer amount of time and would likely sustain more damage as a result. This information (e.g., number of seconds elapsed and units of damage sustained) could be included in the Authenticatable Outcome Message AOM so that the central computer **12** can compare it to known information to determine whether it was within "normal" bounds. If it was outside normal bounds, the central computer **12** may initiate a challenge/response protocol involving the game computer **14**, including certain register values (such as invulnerability status) in the reply message. Rather than detecting the presence of a GAME GENIE directly, this protocol detects the end-effects of a GAME GENIE. Software obfuscation techniques can also be used to effectively hide how the game software **15** works, as is well known in the art. Reverse engineering obfuscated software requires considerable time, delaying the creation of GAME GENIE produced cheat codes. Since CD-ROMs must be re-mastered every ten thousand or so pressings, it is possible to create many different versions of the game software. Thus, a GAME GENIE device would have to generate cheat codes for every possible software variation.

Another solution is to monitor the time interval between the time the game computer **14** loads the game program **26** and the time that the game actually starts. If a GAME GENIE device were being used, the game would not start immediately since the player has to enter the cheat codes into the game computer **14** prior to game start.

In yet another embodiment, storing all the game data on removable memory media may enable the central authority to subsequently determine if the game was created with cheat codes input by a GAME GENIE device. This data may be "recorded" as described in detail herein.

The above described tampering indications may be incorporated into the Authenticatable Outcome Message AOM as "subliminal channels" of information, i.e., information which is difficult to decipher. In addition to the hash value and encryption authentication protocols described above, the game software **15** may run an integrity check on itself consisting of, for example, performing a one-way hash of the current memory registers to obtain a hash value. It then determines whether this hash value is within an allowable range of possible hash values stored as a line of code in the game program. If the determined hash value is within the allowable range, it returns a tamper indication value of 0 (i.e., no tampering made or attempted). If the determined hash value is outside the allowable range, it returns a tamper indication value of 1 (i.e., tampering made or attempted). This tamper indication value 0 or 1 is appended to the outcome and incorporated into the Authenticatable Outcome Message AOM. When the Authenticatable Outcome Message AOM is authenticated by the central computer **12**, the tamper indication digit is interpreted to indicate whether that copy of the game software **15** has been altered or modified. These messages may be arranged so as to render them very difficult for a hacker to interpret their meaning. For example, in the string 13000087457, the last digit "7" is a pointer to the seventh digit in the string—"8", where the fact that this digit is an even number indicates that tampering was attempted. Similarly, the game software **15** may generate scores in specified multiples, e.g., five, such that any score not ending in a five or a zero is invalid. Furthermore, the game software **15** may vary one digit in the score to indicate tampering therewith in accordance with a self-integrity check as described above. For example, a score of 3905 is valid, but if the score is 3905+1=3906, the score is rejected because the addition of the numeral **1** indicates tampering.

The natural random variations in the magnetic memory media on which the game software **15** is made available, may be detected and used as a secret or private key in the cryptographic protocols described herein. These characteristics include variations in coercivity, granularity, coating thickness, surface profile, and the like. Thus, each specimen