



US 20100248816A1

(19) **United States**

(12) **Patent Application Publication**  
**Liber et al.**

(10) **Pub. No.: US 2010/0248816 A1**

(43) **Pub. Date: Sep. 30, 2010**

(54) **VALIDATION SCHEDULING IN A WAGERING GAME MACHINE**

(86) PCT No.: **PCT/US08/05993**

§ 371 (c)(1),  
(2), (4) Date: **Nov. 13, 2009**

(75) Inventors: **Nevin J. Liber**, Libertyville, IL  
(US); **Jason A. Smith**, Vernon Hills,  
IL (US)

**Related U.S. Application Data**

(60) Provisional application No. 60/917,994, filed on May 15, 2007.

Correspondence Address:  
**SCHWEGMAN, LUNDBERG & WOESSNER/  
WMS GAMING  
P.O. BOX 2938  
MINNEAPOLIS, MN 55402 (US)**

**Publication Classification**

(51) **Int. Cl.**  
**A63F 9/24** (2006.01)  
(52) **U.S. Cl.** ..... **463/25**

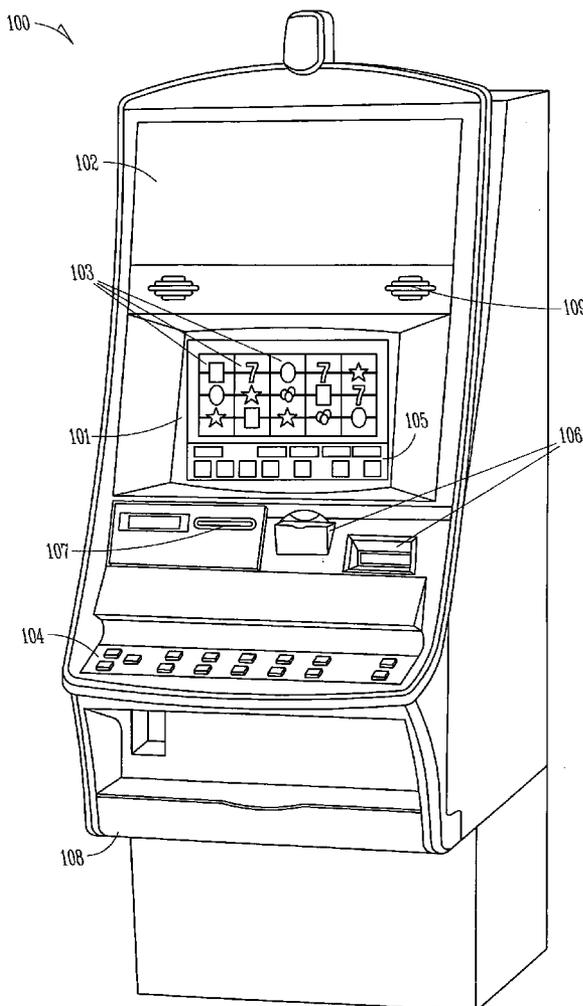
(57) **ABSTRACT**

A computerized wagering game system includes a gaming module comprising gaming code which is operable to present a wagering game on which monetary value can be wagered, and a validation module. The validation module is operable to validate two or more queues of data, and to dynamically adjust validation speed of each queue based on an estimated completion time.

(73) Assignee: **WMS Gaming Inc.**, Waukegan, IL  
(US)

(21) Appl. No.: **12/600,137**

(22) PCT Filed: **May 9, 2008**



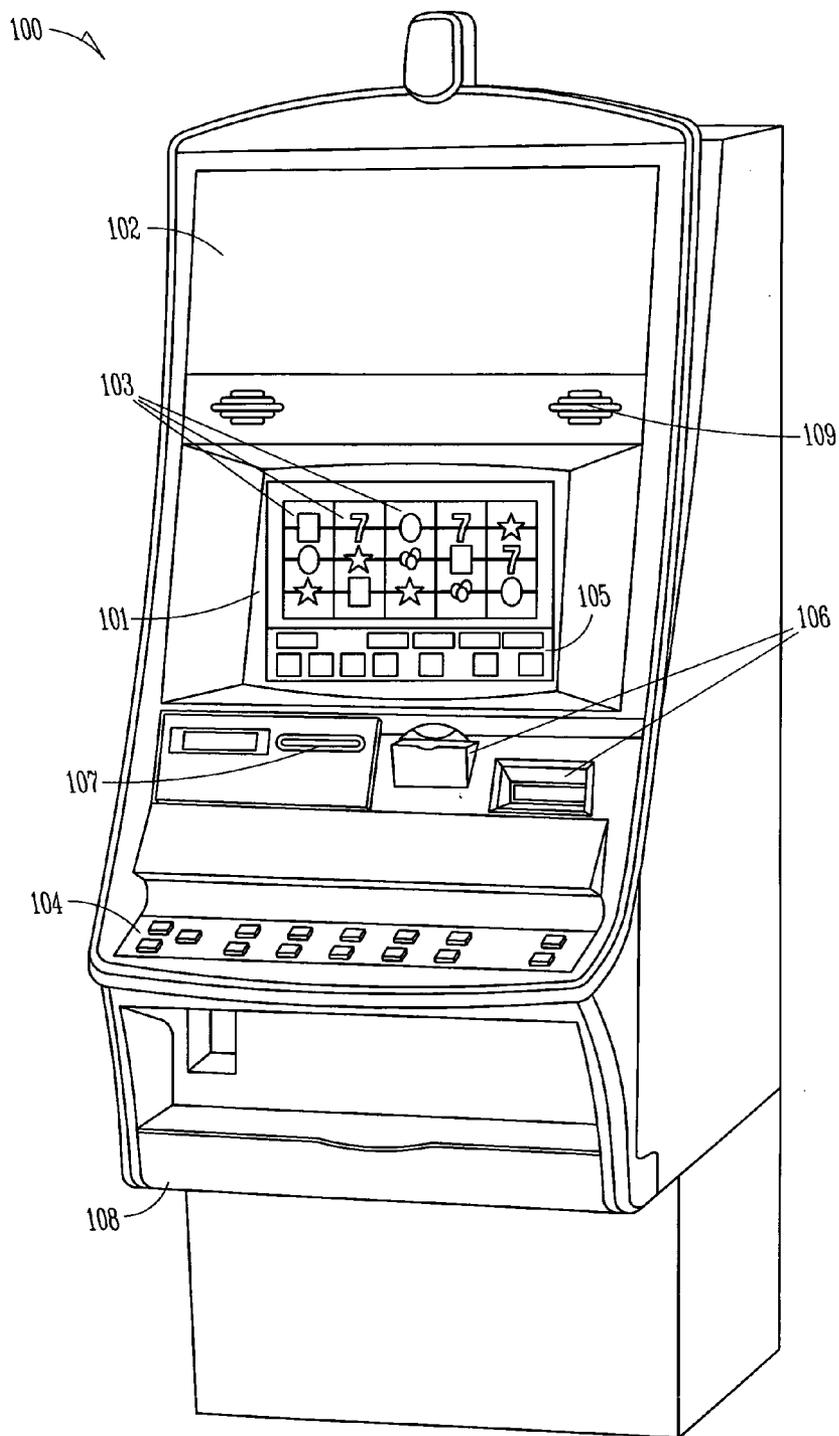


FIG. 1

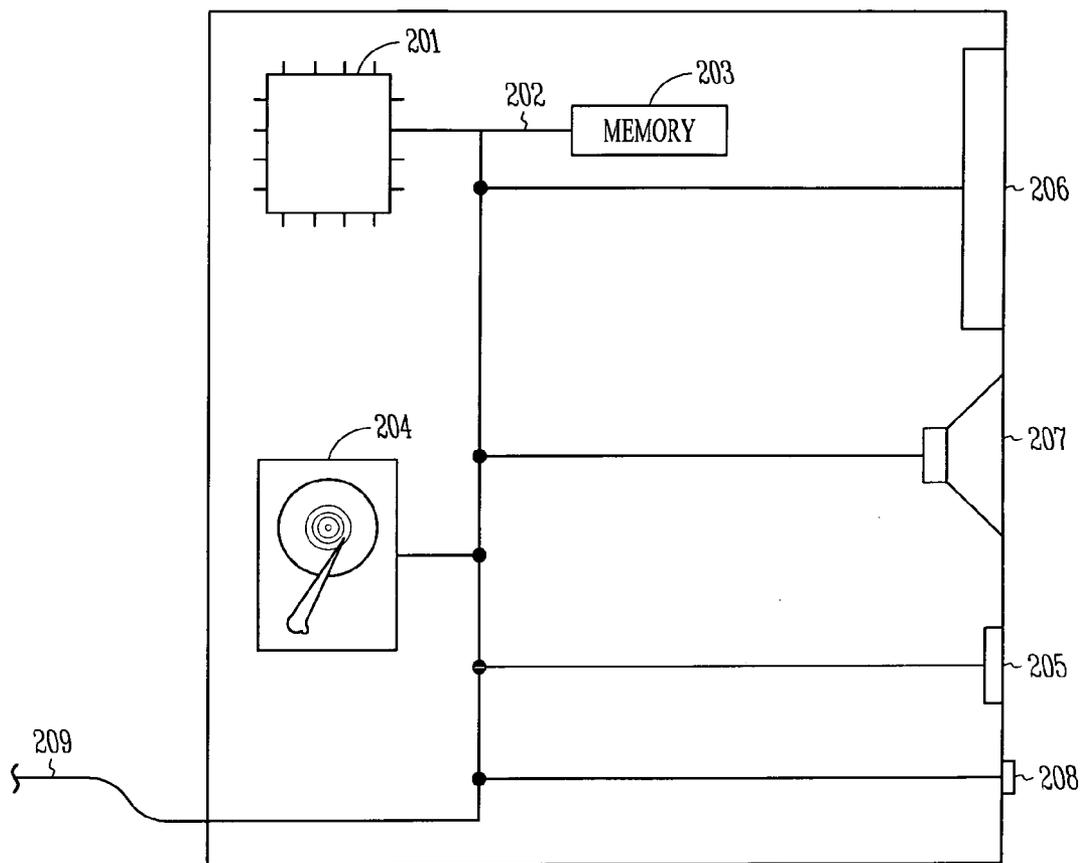


FIG. 2

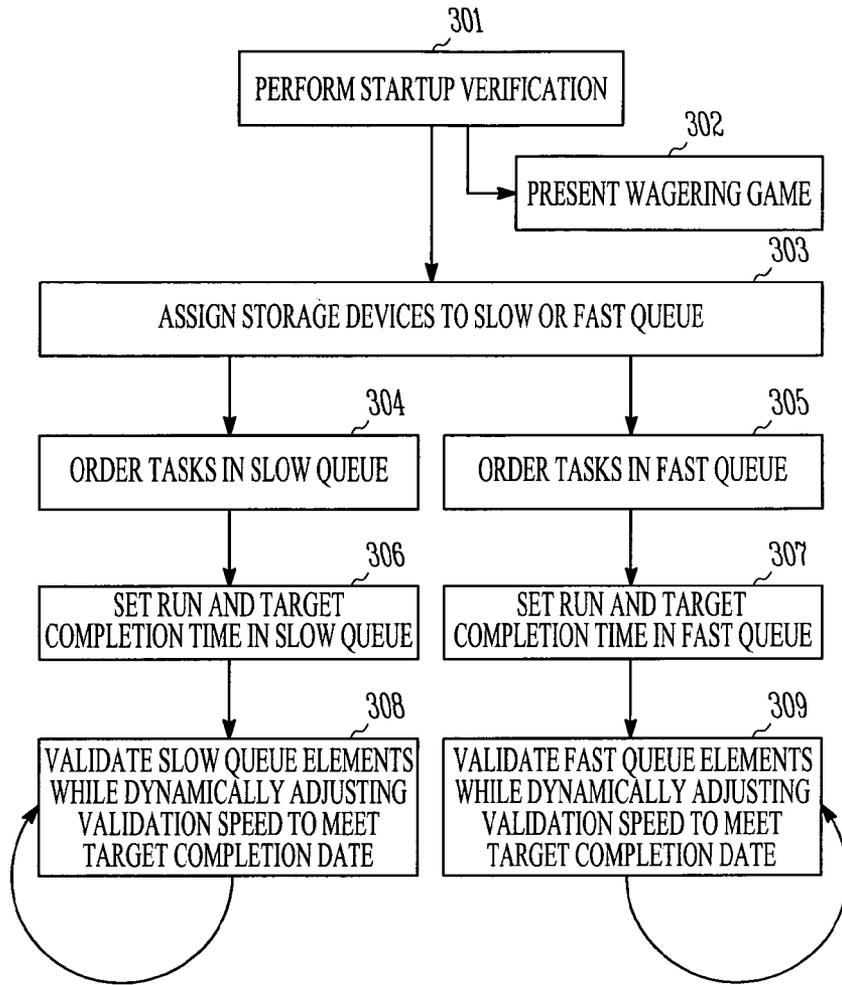


FIG. 3

SLOW DEVICE LIST ~ 401

FAST DEVICE LIST ~ 402

BIOS  
COMPACT FLASH DEVICES  
EEPROM  
OPTICAL DISK

SYSTEM MEMORY  
HARD DISK DRIVE

FIG. 4

**VALIDATION SCHEDULING IN A WAGERING GAME MACHINE**

**RELATED APPLICATION**

[0001] This patent application claims the priority benefit of U.S. Provisional Patent Application Ser. No. 60/917,994 filed May 15, 2007 and entitled "VALIDATION SCHEDULING IN A WAGERING GAME MACHINE", the content of which is incorporated herein by reference in its entirety.

**FIELD OF THE INVENTION**

[0002] The invention relates generally to validating the data in a wagering game machine environment, and more specifically to scheduling validation of data in a wagering game machine.

**LIMITED COPYRIGHT WAIVER**

[0003] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever. Copyright 2007, 2008, WMS Gaming, Inc.

**BACKGROUND**

[0004] Computerized wagering games have largely replaced traditional mechanical wagering game machines such as slot machines, and are rapidly being adopted to implement computerized versions of games that are traditionally played live such as poker and blackjack. These computerized games provide many benefits to the game owner and to the gambler, including greater reliability than can be achieved with a mechanical game or human dealer, more variety, sound, and animation in presentation of a game, and a lower overall cost of production and management.

[0005] The elements of computerized wagering game systems are in many ways the same as the elements in the mechanical and table game counterparts in that they must be fair, they must provide sufficient feedback to the game player to make the game fun to play, and they must meet a variety of gaming regulations to ensure that both the machine owner and gamer are honest and fairly treated in implementing the game. Further, they must provide a gaming experience that is at least as attractive as the older mechanical gaming machine experience to the gamer, to ensure success in a competitive gaming market.

[0006] Computerized wagering games do not rely on the dealer or other game players to facilitate game play and to provide an entertaining game playing environment, but rely upon the presentation of the game and environment generated by the wagering game machine itself. Incorporation of audio, video, and mechanical features into wagering game systems enhance the environment presented are therefore important elements in the attractiveness and commercial success of a computerized wagering game system. A variety of complex graphics and video capabilities are also often provided via one or more specialized graphics processors, including the ability to decode and render full motion video, and to render complex three-dimensional graphics.

[0007] The software programs that control presentation of the wagering game are typically stored on a nonvolatile storage device such as a hard disk drive or flash memory, and are

desirably secure. In many wagering game machines, the software in the machine determines whether the player wins or not in a given round of game play, so integrity of the wagering game machine software is important to ensuring fair and reliable game play. The software programs in the wagering game machine are therefore typically validated when the machine starts and while the machine runs, ensuring that it has not been altered or damaged.

[0008] It is therefore desirable to manage validation of software in a wagering game machine.

**SUMMARY**

[0009] One example embodiment of the invention comprises a computerized wagering game system including a gaming module comprising gaming code which is operable when executed on to conduct a wagering game on which monetary value can be wagered, and a validation module. The validation module is operable in various embodiments to validate two or more queues of data, and to dynamically adjust validation speed of each queue based on an estimated completion time.

**BRIEF DESCRIPTION OF THE FIGURES**

[0010] FIG. 1 shows a computerized wagering game machine, as may be used to practice some example embodiments of the invention.

[0011] FIG. 2 is a block diagram of a wagering game system, consistent with some example embodiments of the invention.

[0012] FIG. 3 illustrates a method of continuously validating devices or data sets within a wagering game machine using multiple queues and dynamic validation rates, consistent with an example embodiment of the invention.

[0013] FIG. 4 illustrates a variety of device types assigned to either a fast device queue or a slow device queue, consistent with an example embodiment of the invention.

**DETAILED DESCRIPTION**

[0014] In the following detailed description of example embodiments of the invention, reference is made to specific examples by way of drawings and illustrations. These examples are described in sufficient detail to enable those skilled in the art to practice the invention, and serve to illustrate how the invention may be applied to various purposes or embodiments. Other embodiments of the invention exist and are within the scope of the invention, and logical, mechanical, electrical, and other changes may be made without departing from the subject or scope of the present invention. Features or limitations of various embodiments of the invention described herein, however essential to the example embodiments in which they are incorporated, do not limit the invention as a whole, and any reference to the invention, its elements, operation, and application do not limit the invention as a whole but serve only to define these example embodiments. The following detailed description does not, therefore, limit the scope of the invention, which is defined only by the appended claims.

[0015] One example embodiment of the invention comprises a computerized wagering game system including a gaming module comprising gaming code which is operable when executed on to conduct a wagering game on which monetary value can be wagered. The wagering game system also comprises an authentication module operable to authen-

ticate one or more used partitions of a nonvolatile storage volume using a first authentication method and operable to authenticate or validate one or more unused sections of the nonvolatile storage volume using a second authentication method. In some embodiments, this enables use of a variety of different storage volume sizes while keeping the used space and the used space's authentication data the same, and enables faster authentication of the unused space on the nonvolatile storage volume.

[0016] FIG. 1 illustrates a computerized wagering game machine, as may be used to practice some embodiments of the present invention. The computerized gaming system shown generally at 100 is a video wagering game system, which displays information for at least one wagering game upon which monetary value can be wagered on video display 101. In a further example, a second video display 102 is provided as a part of a top-box assembly, such as to display a bonus game or other information. Video displays 101 and 102 are in various embodiments a CRT display, a plasma display, an LCD display, a surface conducting electron emitter display, or any other type of display suitable for displaying electronically provided display information. Alternate embodiments of the invention will have other game indicators, such as mechanical reels instead of the video graphics reels shown at 103 that comprise a part of a video slot machine wagering game.

[0017] A wagering game is presented using software within the wagering game machine, such as through instructions stored on a machine-readable medium such as a hard disk drive or nonvolatile memory. In some further example embodiments, some or all of the software stored in the wagering game machine is encrypted or is verified using a hash algorithm or encryption algorithm to ensure its authenticity and to verify that it has not been altered. For example, in one embodiment the wagering game software is loaded from nonvolatile memory in a compact flash card, and a hash value is calculated or a digital signature is derived to confirm that the data stored on the compact flash card has not been altered. The game of chance implemented via the loaded software takes various forms in different wagering game machines, including such well-known wagering games as reel slots, video poker, blackjack, craps, roulette, or hold 'em games. The wagering game is played and controlled with inputs such as various buttons 104 or via touchscreen overlay buttons 105 on video screen 101. In some alternate examples, other devices such as pull arm are used to initiate reel spin in this reel slot machine example are employed to provide other input interfaces to the game player.

[0018] Monetary value is typically wagered on the outcome of the games, such as with tokens, coins, bills, or cards that hold monetary value. The wagered value is conveyed to the machine through a changer 106 or a secure user identification module interface 107, and winnings are returned via the returned value card or through the coin tray 108. Sound is also provided through speakers 109, typically including audio indicators of game play, such as reel spins, credit bang-ups, and environmental or other sound effects or music to provide entertainment consistent with a theme of the computerized wagering game. In some further embodiments, the wagering game machine is coupled to a network, and is operable to use its network connection to receive wagering game data, track players and monetary value associated with a player, and to perform other such functions.

[0019] FIG. 2 shows a block diagram of an example embodiment of a Wagering game system. The wagering game system includes a processor 201, which is sometimes called a microprocessor, controller, or central processing unit (CPU). In some embodiments, more than one processor is present, or different types of processors are present in the wagering game system, such as using multiple processors to run gaming code, or using dedicated processors for audio, graphics, security, or other functions. The processor is coupled via a bus 202 to various other components, including memory 203 and nonvolatile storage 204. The nonvolatile storage is able to retain the data stored therein when power is removed, and in various embodiments takes the form of a hard disk drive, nonvolatile random access memory such as a compact flash card, or network-coupled storage. Further embodiments include additional data storage technologies, such as compact disc, DVD, or HD-DVD storage in the wagering game system.

[0020] The bus 202 also couples the processor and components to various other components, such as a value acceptor 205, which is in some embodiments a token acceptor, a card reader, or a biometric or wireless player identification reader. A touchscreen display 206 and speakers 207 serve to provide an interface between the wagering game system and a wagering game player, as do various other components such as buttons 208, pullarms, and joysticks. A network connection 209 couples the wagering game system to other wagering game machines and to a wagering game server, such as to provide downloadable games or to provide accounting, player tracking, or other functions. These components are located in a wagering game machine cabinet such as that of FIG. 1 in some embodiments, but can be located in multiple enclosures comprising a wagering game system or outside a wagering game machine cabinet in other embodiments, or in alternate forms such as a wireless or mobile device.

[0021] In operation, the wagering game system loads program code from nonvolatile storage 204 into memory 203, and the processor 201 executes the program code to cause the wagering game system to perform desired functions such as to present a wagering game upon which monetary value can be wagered. This and other functions are provided by various modules in the computerized system such as an audio module, a game presentation module, or a touchscreen display module, where such modules comprise in some embodiments hardware, software, mechanical elements, manual intervention, and various combinations thereof. The wagering game machine is coupled to other wagering game machines, and to various other elements such as game servers, accounting servers, or community or progressive game servers via the network connection 209, and exchanges data with these machines via the network connection.

[0022] The nonvolatile storage 204 is in some embodiments a hard disk drive, flash memory, or another nonvolatile storage device or group of devices having one or more partitions. In one example, a single hard drive is split into three separate partitions, each of which is addressable and can be managed as though it were a separate storage device. In another example, a nonvolatile flash memory comprises a single partition, or a hard disk drive comprises a single partition.

[0023] Multiple storage volumes or multiple partitions are used in some embodiments for purposes including using different partitions to store operating system code, wagering game code, multimedia information, and downloaded infor-

mation such as downloadable games. Further, some operating systems such as Linux can benefit from using a separate partition or separate nonvolatile storage device for virtual memory, by which a computerized wagering game machine can store “pages” of information not currently being used in memory in nonvolatile storage such as on a hard disk drive, freeing up main memory for other data. In other examples, one or more partitions are used to store the operating system, wagering game code, and other data that should be kept secure to ensure fair and reliable operation of the wagering game, while other partitions store less sensitive data such as sound and graphics files, or store downloaded game code not yet verified and installed onto a secure partition for execution.

**[0024]** The data stored on the wagering game machine is verified to ensure that the game presented is fair, and to ensure that the game code has not been altered since it was manufactured. If the data in the wagering game system was not authenticated, a cheat could alter the rules of play or change the odds by altering the game code, resulting in significant loss to the game operator. Verification of the partition or volume information for the nonvolatile storage volume **204** relies in some embodiments on cryptographic technology such as digital signatures or certificates. These encryption technologies typically utilize a symmetric or asymmetric algorithm, designed to obscure the data such that a specific key is needed to read or alter the data. A symmetric algorithm relies on agreement of a secret key before encryption, and the decryption key is either the same as or can be derived from the encryption key. Secrecy of the key or keys is vital to ensuring secrecy of the data in such systems, and the key must be securely distributed to the receivers before decryption such as via a secure key exchange protocol. Common symmetric algorithms include DES, 3DES or triple-DES, AES, Blowfish, Twofish, IDEA, RD2, RC4, and RC5.

**[0025]** Public key algorithms, or asymmetric algorithms, are designed so that the decryption key is different than and not easily derivable from the encryption key. The term “public key” is used because the encryption key can be made public without compromising the security of data encrypted with the encryption key. Anyone can therefore use the public key to encrypt a message, but only a receiver with the corresponding decryption key can decrypt the encoded data. The encryption key is often called the public key, and the decryption key is often called the private key in such systems.

**[0026]** Such systems can be used to digitally sign a document where the signer uses the secret private key to encrypt the document or some portion of it such as a one-way hash of the document, and then publishes the encrypted message. Anyone can use the signer’s published or known public key to decrypt the signed message, confirming that it was encrypted or signed by the owner of the public/private key pair. In some examples, the publisher of a wagering game executable, an operating system, or other partition contents digitally signs the contents of the partition such that the partition can be verified by decrypting the partition or a signed hash of the partition with a known and trusted public key. Common public key algorithms include RSA, elliptic curve cryptography, Diffie-Hellman, and ElGamal.

**[0027]** One-way hash functions take an input string and derive a fixed length hash value. The hash value is typically of significantly shorter length than the document, and is often calculated by application of some type of data compression algorithm. The functions are designed so that it is extremely difficult to produce an input string that produces a certain

hash value, resulting in a function that is considered one-way. A particular set of data such as a document, software program, or storage volume can therefore be checked for authenticity by verifying that the hash value resulting from a given one-way hash function is the same as what was computed using a known-good copy of the data, making authentication of data relatively certain. Hash functions can be combined with other methods of encryption or addition of secret strings of text in the input string to ensure that only the intended parties can encrypt or verify data using the one-way hash functions. Common examples of one-way hash function encryption include MD2, MDC2, MD4, MD5, and SHA.

**[0028]** A variation on one-way hash functions is use of Message Authentication Codes, or MAC. A MAC comprises a one-way hash function that further includes a secret key, such that knowledge of the key is necessary to encode or verify a given set of data. MACs are particularly useful where the hash value would otherwise be subject to unauthorized alteration or replacement, such as when transmitted over a public network or a network that would be difficult to protect, such as a very large network linking hundreds of computerized wagering game machines in a large gaming facility.

**[0029]** Similarly, a digital signature can be applied to a volume of data, or to a one-way hash derived from a volume of data, to ensure that the hash value being used for comparison is authentic. For example, a wagering game manufacturer may sign a has value for a downloadable game with a private key, such that the wagering game system downloading the game decrypts the encrypted hash value with the wagering game manufacturer’s public key before the hash value can be used to verify or authenticate the downloaded game. This ensures the authenticity of the hash value, which can then be used to authenticate or validate the downloaded game code.

**[0030]** But, application of verification technologies such as digital signatures and hash value calculations are time consuming, and can have a significant impact on the input/output and processing resources of a wagering game machine. Further, some regulations require that the entire volume on which a secure partition is stored must be verified both at startup and periodically during operation, such that verification processes are scheduled to consume a significant portion of the resources available to a running wagering game machine.

**[0031]** Some embodiments of the invention provide a validation module within the wagering game system that is operable in various examples to validate two or more queues of data, and to dynamically adjust validation speed of each queue based on an estimated completion time. One such validation module is illustrated in FIG. 3, which is a flowchart illustrating an example method of validating multiple sets of data using different validation queues that are independently dynamically scheduled based on an estimated completion time.

**[0032]** At **301**, the wagering game machine is started, and an initial startup verification is performed. In some embodiments, the BIOS, a preboot execution environment program (PXE boot program), or another element of the wagering game machine verifies the content of various nonvolatile storage devices such as hard disk drives, compact flash drives, and the BIOS or other operating code before the wagering game machine becomes operable to present a wagering game upon which monetary value can be wagered. Once this initial verification is completed, the wagering game machine presents the wagering game at **302**, and proceeds to perform periodic

revalidation of various nonvolatile storage devices as well as verification of data stored in memory.

**[0033]** Each device to be verified on an ongoing basis is assigned to a particular queue at **303**, which in this example include a slow device queue and a fast device queue that ran at the same time, independent from one another. The slow device queue in this example includes devices that have relatively slow data transfer rates or that consume a relatively significant amount of wagering game machine resources to operate, such as Compact Flash storage volumes or EEPROM devices such as the wagering game system BIOS. The fast device queue includes devices that are relatively fast, such as the main memory of the wagering game system, including program data loaded from nonvolatile storage to present the wagering game to a game player.

**[0034]** The devices or elements placed in each queue are ordered in some embodiments. The tasks in the slow queue are ordered at **304**, and the tasks in the fast queue are ordered at **305**, although in alternate embodiments the actual order of verification is not important and the order is assigned arbitrarily. Here, the order can be specified or changed by the game programmer or by the game's owner or operator, to ensure compliance with gaming regulations or for other reasons. Once the various devices or elements to be verified are assigned to one of the slow or fast queues, start or run times are set, along with target completion times.

**[0035]** At **306**, the run time and the target completion time is set for the slow queue. For example, the owner/operator of the wagering game machine may configure the wagering game system to start validation at 7 o'clock A.M., with a target completion time set at 7 o'clock A.M, so that the wagering game machine completes validation of each element in the queue approximately once every day. In another example, the tasks in the fast queue have a run time set at **307** of 7 o'clock A.M., with target completion times of 7 o'clock P.M. and 7 o'clock A.M., indicating that the fast queue will complete validation of the elements in its queue approximately every twelve hours. The fast queue therefore remains inactive during peak casino hours of 7 o'clock P.M. to 7 o'clock A.M., and every element in the queue is validated once every 24 hours. In another example, the target completion time is specified as a certain amount of time after the queue start time, such as 12 hours after validation begins.

**[0036]** The validation of elements then takes place for the slow queue at **308**, and the fast queue at **309**. The validation process can be triggered by calendar or queue, or can be triggered by a software message or a command line interface instruction requesting validation of a device. In some embodiments, devices are placed into the queues and are ordered in the queues by using similar command line instructions or software messages passed to the validation module. Validation takes place via an independent operation for each queue, such that the validation rate of the slow queue is managed separately from the rate of validation of the fast queue.

**[0037]** As shown in FIG. 3, validation of both the slow queue at **308** and the fast queue at **309** occurs in parallel, and repeats over time such that the data is being continuously and repeatedly validated as the wagering game machine operates over a period of days, weeks, or even months or more of continuous operation. Each of the two validation operations has the run and target completion times such as were set at **306** and **307**, and is able to dynamically estimate how much work is needed to complete validation of all elements in the queue,

and to adjust the priority or speed at which the validation process occurs so that the process finishes before the target completion time. This may be necessary in some jurisdictions to meet gaming regulations for validation of certain devices or data sets, while in other embodiments the target completion time can be considered a desired or estimated completion time such that completion at slightly after the desired completion time is allowable.

**[0038]** The dynamic adjustment of validation is based in various embodiments on a number of factors, including historic completion time for various devices or data sets in the queue, observed volume of data validated per period of time at various validation process priority settings, and other such factors to more accurately estimate the amount of time needed to validate queue elements at various validation rates.

**[0039]** Validation rates are also varied in a further embodiment based on a desired duty cycle or operational time of a device, such as varying the validation rate of a hard disk drive to ensure that the hard disk drive is active for no more than 50% of the queue completion time. This is desirable in some embodiments both because less operating power is consumed if the hard disk drive can be shut down or will be known to be inactive for some periods of time, but also because many hard disk drives are rated for mean time between failure (MTBF) assuming operation less than 50% of the time. If the hard disk drive is operated more than the specified 50% of the time, its longevity may be significantly reduced, and the mean operating time between failure can be significantly lower. Similarly, optical disk, compact flash, and other devices can be operated using altered duty cycle or desired operational time criteria, both to reduce power consumption such as where a compact flash memory circuit can be powered down, and to reduce mechanical wear and enhance longevity of the device as with the optical disk drive.

**[0040]** FIG. 4 shows an example device list of devices in a slow device queue and a fast device queue, consistent with an example embodiment of the invention. At **401**, the slow device queue includes devices that are either relatively slow, or are relatively expensive in terms of system resources to operate. Examples include EEPROM devices, which are electrically erasable and programmable memories, such as are used to store system BIOS information or other data. EEPROM devices differ from typical system memory in that the data stored in an EEPROM is retained even when power is not applied. This makes it desirable for use in storing information such as the boot instructions and configuration information stored in the BIOS, so that the information is retained when the wagering game machine is unplugged or turned off. Compact Flash devices operate in a similar way to EEPROM memory, and are commonly found in digital cameras and digital music players to store relatively large volumes of information. Several gigabytes can be stored on a Compact Flash device, making them suitable for use in providing wagering game program code, operating system code, and other such information to wagering game systems. Similarly, optical drives such as CD, DVD, Blu-Ray, and other read-only or read-write optical drives can be used to store large amounts of data for relatively low cost, although the speed of most optical drives is relatively slow.

**[0041]** Fast devices shown at **402** include devices that operate relatively quickly, or that have relatively little impact on the wagering game system to operate or manage. Examples include dynamic system memory, which can be read very

quickly, and hard disk drives, which provide relatively fast storage and retrieval of very large volumes of information.

[0042] These device types and others are used to store a wide variety of data in a wagering game system, much of which should be validated continuously to ensure the integrity of the wagering game system. Various embodiments of the invention presented here provide the ability to validate two or more queues of data, and to dynamically adjust validation speed of each queue based on an estimated completion time. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the example embodiments of the invention described herein. It is intended that this invention be limited only by the claims, and the full scope of equivalents thereof.

1. A computerized wagering game system, comprising: a wagering game module operable to present a wagering game upon which monetary value can be wagered; and a validation module operable to validate data stored within the wagering game machine via two or more independent validation queues, such that each of the two or more queues is operable to concurrently and independently validate queued data, and where each of the two or more queues is further operable to queue two or more data elements for validation.

2. The computerized wagering game system of claim 1, wherein at least two of the two or more validation queues are associated with different types of data storage media.

3. The computerized wagering game system of claim 1, wherein at least two of the two or more validation queues are associated with different data types.

4. The computerized wagering game system of claim 1, wherein at least two of the two or more validation queues have independently determined target completion times.

5. The computerized wagering game system of claim 1, wherein a validation speed of at least one of the two or more validation queues is dynamically adjustable to provide for completion of the data at approximately a target completion time.

6. The computerized wagering game system of claim 1, the validation module further operable to use statistical validation completion times to estimate the relationship between the validation speed and completion time of a queue.

7. A method of operating a computerized wagering game system, comprising:

using the computerized wagering game system, validating data stored within the wagering game machine via two or more independent validation queues, such that each of the two or more queues is operable to concurrently and independently validate queued data, and where each of the two or more queues is further operable to queue two or more data elements for validation.

8. The method of operating a computerized wagering game system of claim 7, wherein at least two of the two or more validation queues are associated with different types of data storage media.

9. The method of operating a computerized wagering game system of claim 7, wherein at least two of the two or more validation queues are associated with different data types.

10. The method of operating a computerized wagering game system of claim 7, wherein at least two of the two or more validation queues have independently determined target completion times.

11. The method of operating a computerized wagering game system of claim 7, wherein a validation speed of at least one of the two or more validation queues is dynamically adjustable to provide for completion of the data at approximately a target completion time.

12. The method of operating a computerized wagering game system of claim 7, further comprising using statistical validation completion times to estimate the relationship between the validation speed and completion time of a queue.

13. A computerized wagering game system, comprising: a wagering game module operable to present a wagering game upon which monetary value can be wagered; and a validation module operable to validate software stored within the wagering game machine, and further operable to dynamically adjust a validation speed based on a target validation completion time.

14. The computerized wagering game system of claim 13, wherein the validation speed is dynamically adjusted to ensure completion of the validation before the target validation completion time.

15. The computerized wagering game system of claim 13, wherein the validation speed is dynamically adjusted to ensure completion of the validation at approximately the target validation completion time.

16. The computerized wagering game system of claim 13, wherein a validation speed is dynamically adjusted independently for each of two or more separate validation queues.

17. The computerized wagering game system of claim 13, wherein the validation speed is dynamically adjusted to manage the duty cycle of a hard disk drive containing data being validated.

18. A method of operating a computerized wagering game system, comprising:

using a processor of the computerized wagering game system to: validate software stored within the computerized wagering game system; and dynamically adjust a validation speed based on a target validation completion time.

19. The method of operating a computerized wagering game system of claim 18, wherein the validation speed is dynamically adjusted to ensure completion of the validation before the target validation completion time.

20. The method of operating a computerized wagering game system of claim 18, wherein the validation speed is dynamically adjusted to ensure completion of the validation at approximately the target validation completion time.

21. The method of operating a computerized wagering game system of claim 18, wherein the validation speed is dynamically adjusted independently for each of two or more separate validation queues.

22. The method of operating a computerized wagering game system of claim 18, wherein the validation speed is dynamically adjusted to manage the duty cycle of a hard disk drive containing an image being validated.

23. A machine-readable medium comprising instructions, which when executed by a machine, cause the machine to:

validate data stored within the machine via two or more independent validation queues, such that each of the two or more queues is operable to concurrently and independently validate queued data, and where each of the two or more queues is further operable to queue two or more data elements for validation.