



US 20080220879A1

(19) **United States**(12) **Patent Application Publication**
Barrie et al.(10) **Pub. No.: US 2008/0220879 A1**(43) **Pub. Date: Sep. 11, 2008**(54) **TRUSTED CABINET IDENTIFICATION METHOD**

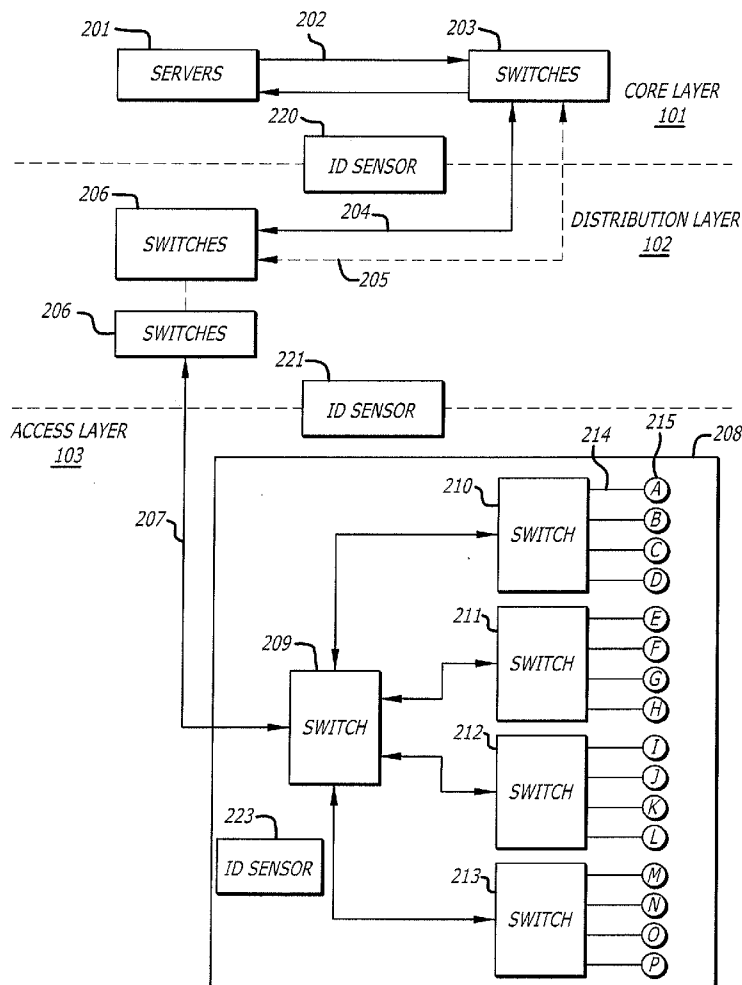
of application No. 11/220,781, filed on Sep. 7, 2005, Continuation-in-part of application No. 11/550,781, filed on Oct. 18, 2006.

(75) Inventors: **Fred Barrie**, Sparks, NV (US);
Harold K. Robb, Reno, NV (US)**Publication Classification**(51) **Int. Cl.**
A63F 9/24 (2006.01)
(52) **U.S. Cl.** **463/42**Correspondence Address:
STEPTOE & JOHNSON, LLP
2121 AVENUE OF THE STARS, SUITE 2800
LOS ANGELES, CA 90067 (US)(57) **ABSTRACT**

A method of identifying components within a gaming machine cabinet that is linked to a central server over a network is disclosed in which the gaming machine includes a plurality of components to be identified within the cabinet. The method includes storing a unique cabinet identifier and encryption key; generating a digital signature from the encryption key; transmitting the unique cabinet identifier to a component to be identified within the cabinet upon power-up of the component; transmitting the digital signature to the component to confirm origin of the unique cabinet identifier; and enabling components within the gaming machine cabinet to register with a game floor manager system on the central server via the network.

(73) Assignee: **BALLY GAMING, INC.**, Las Vegas, NV (US)(21) Appl. No.: **12/109,013**(22) Filed: **Apr. 24, 2008****Related U.S. Application Data**

(63) Continuation-in-part of application No. 11/319,034, filed on Dec. 23, 2005, which is a continuation-in-part



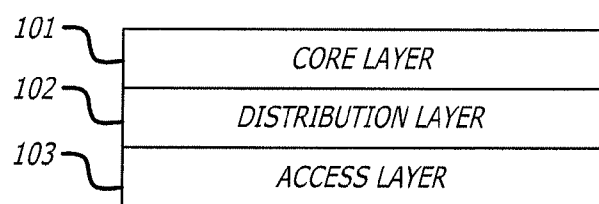


FIG. 1

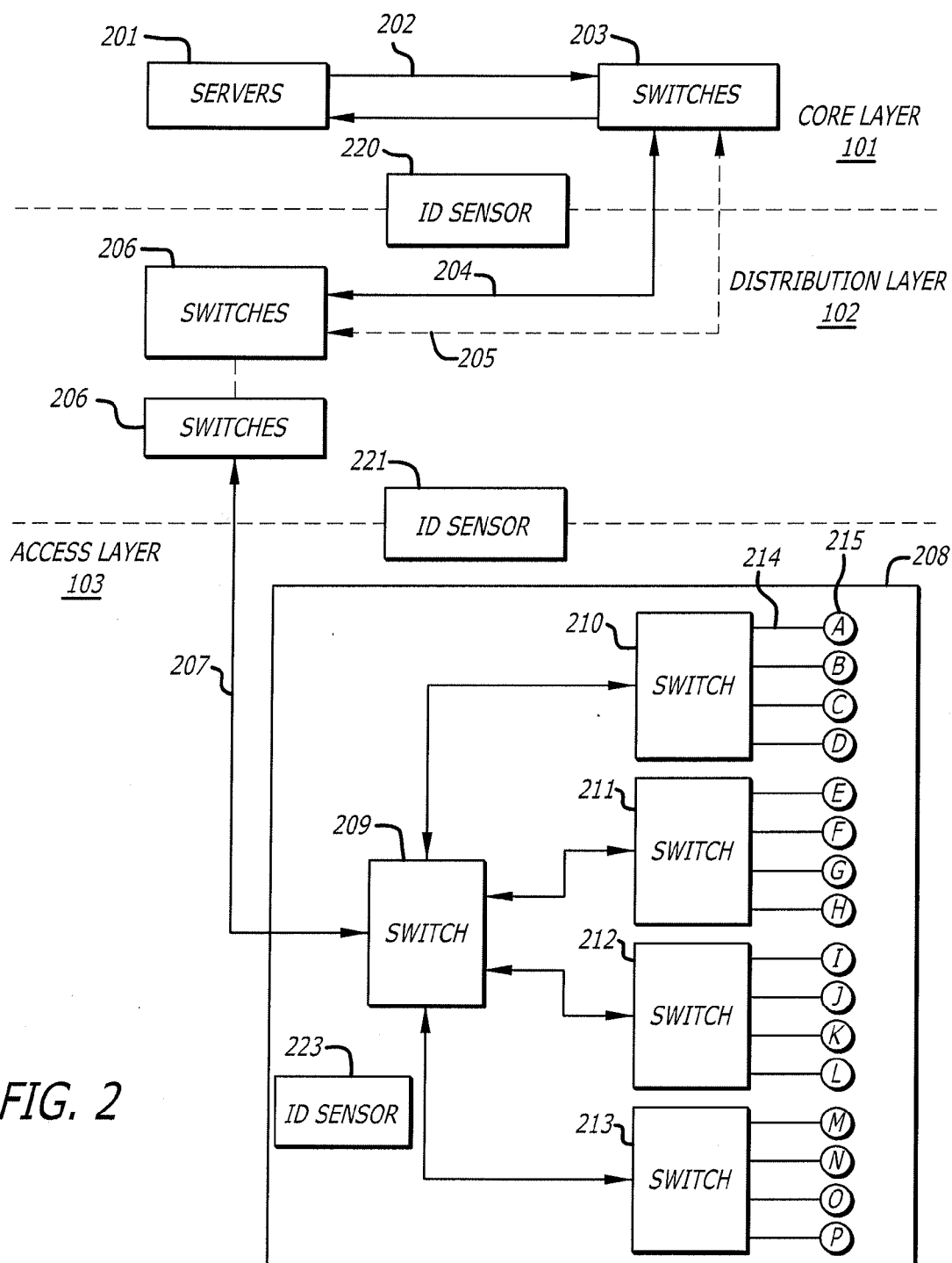


FIG. 2

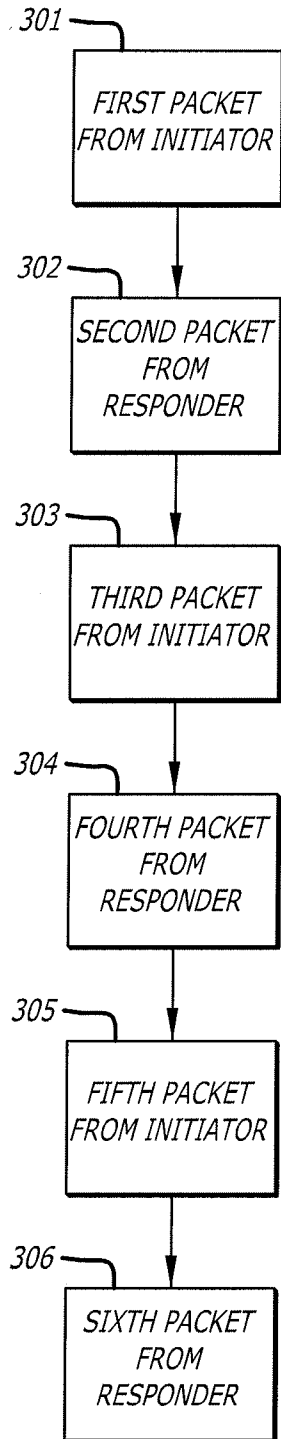


FIG. 3

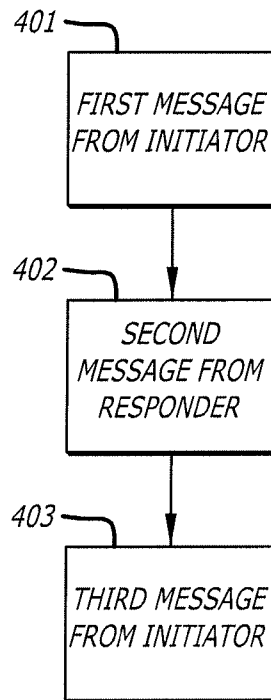


FIG. 4

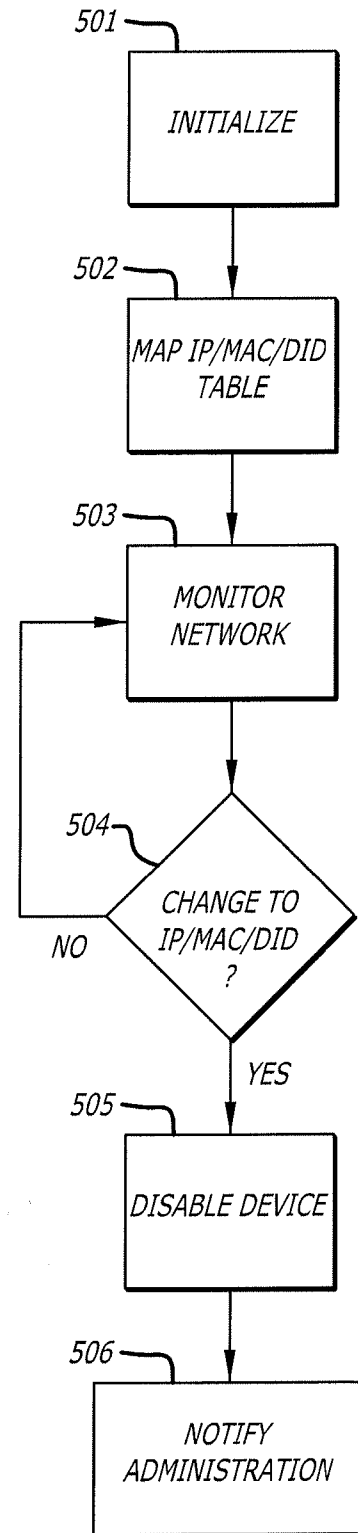


FIG. 5

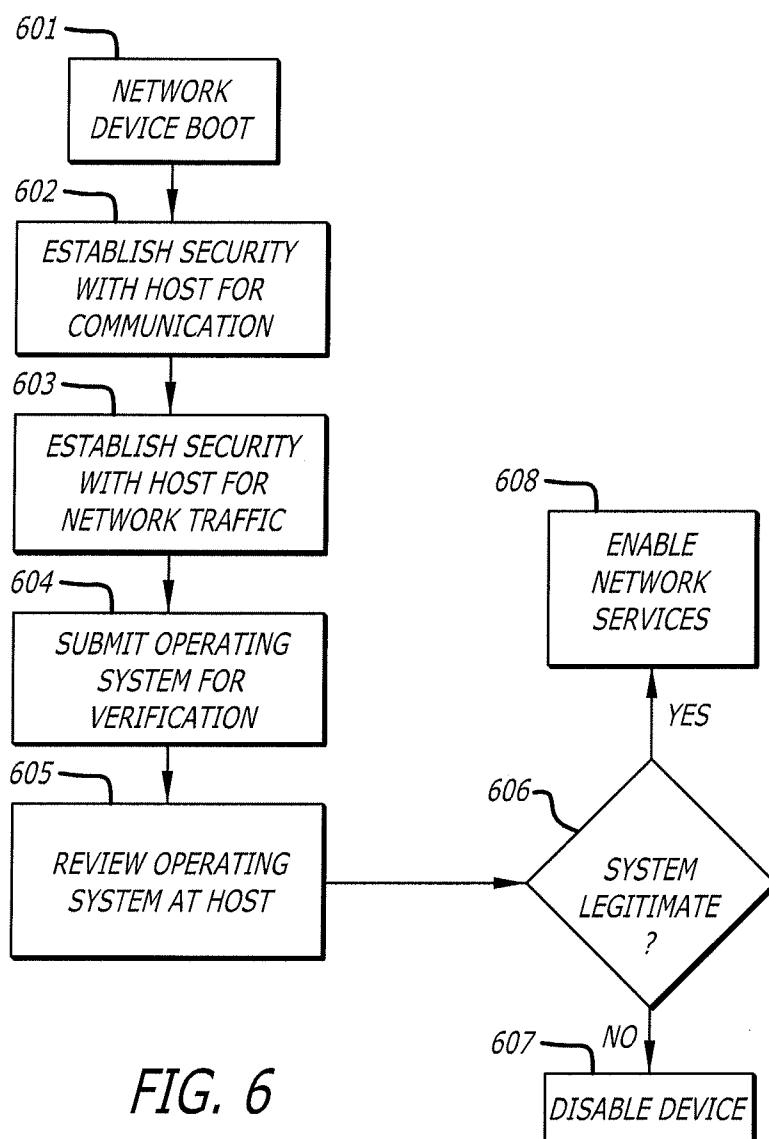


FIG. 6

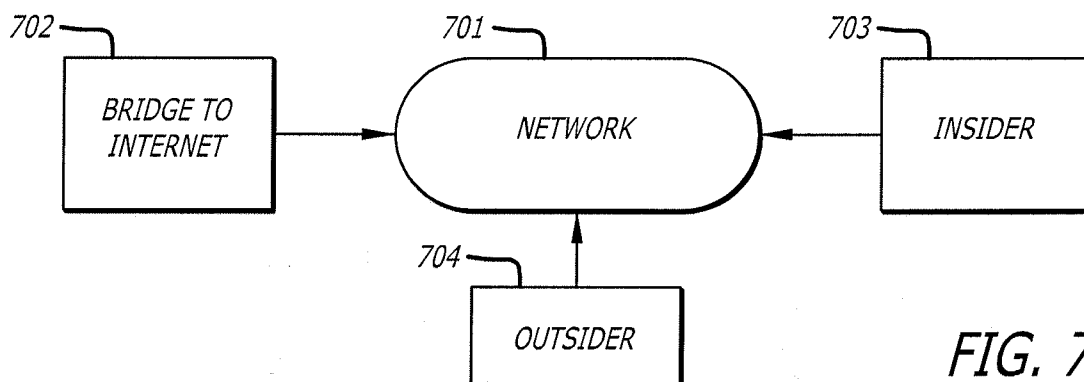


FIG. 7

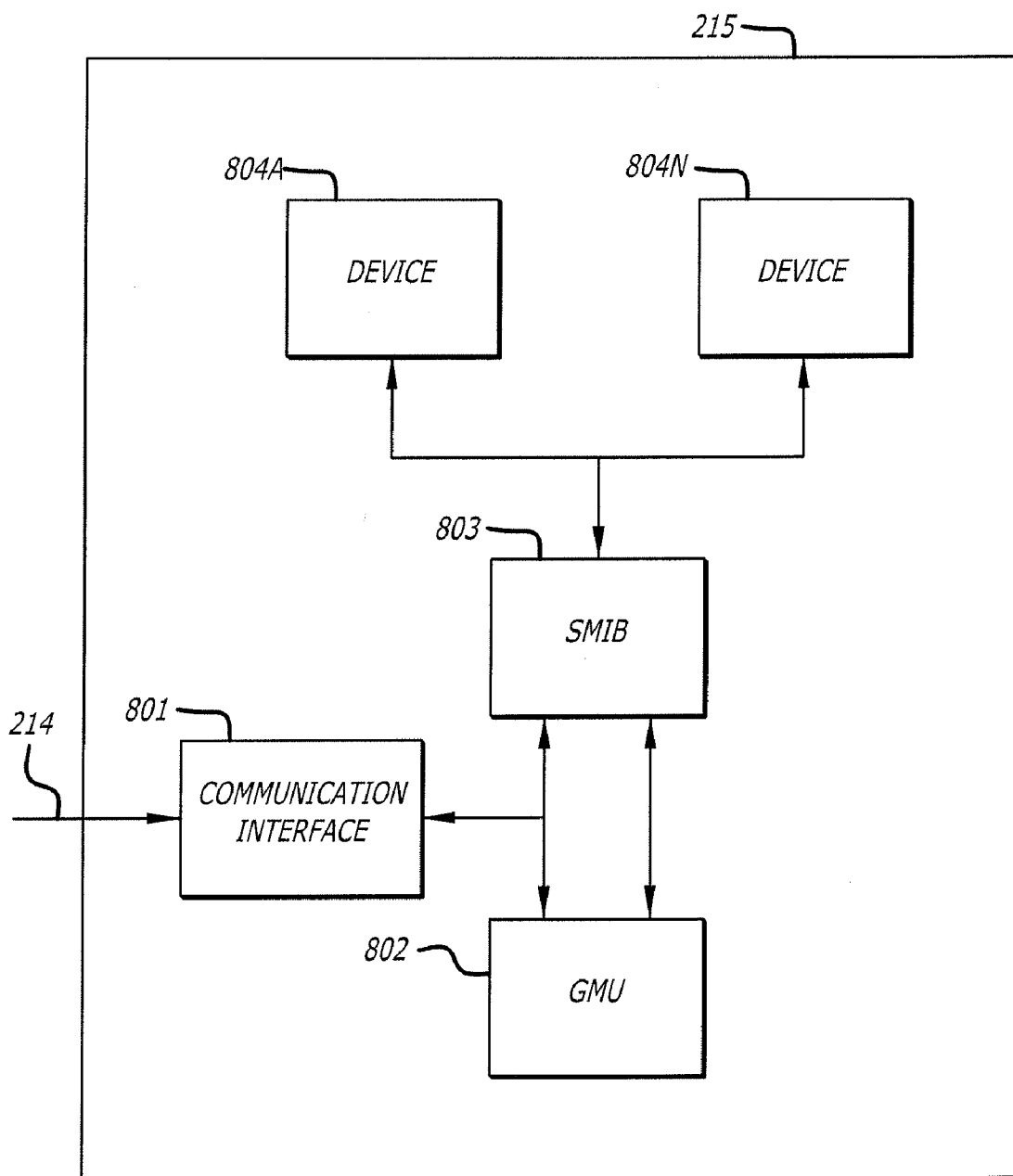


FIG. 8

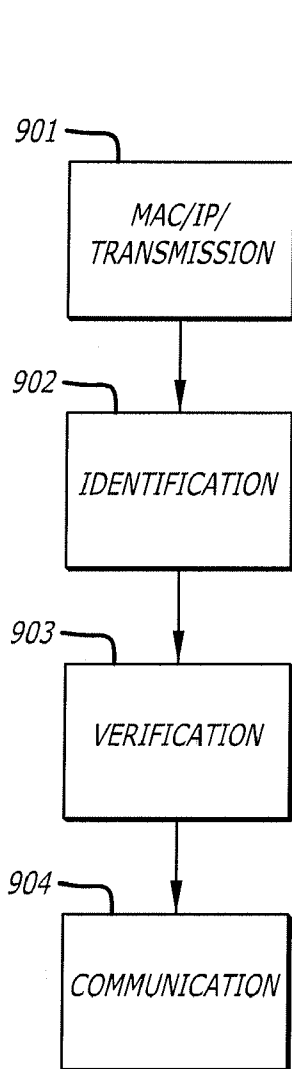


FIG. 9

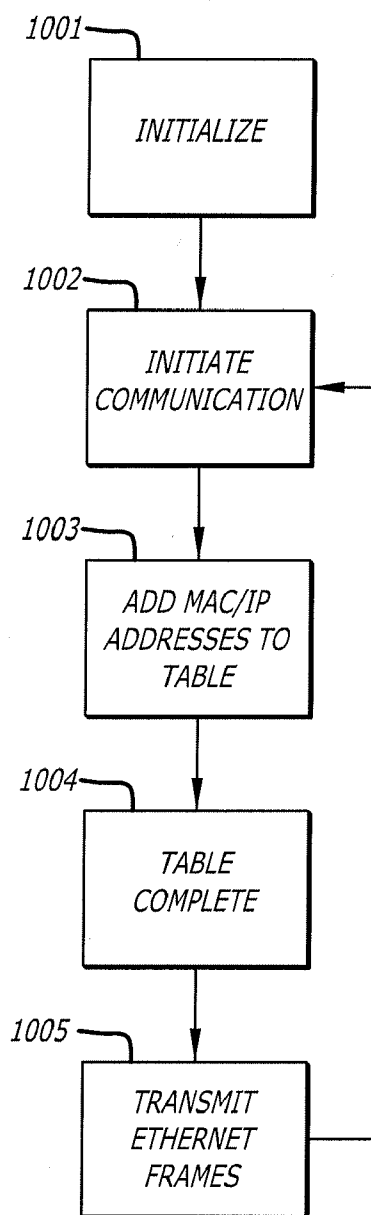


FIG. 10

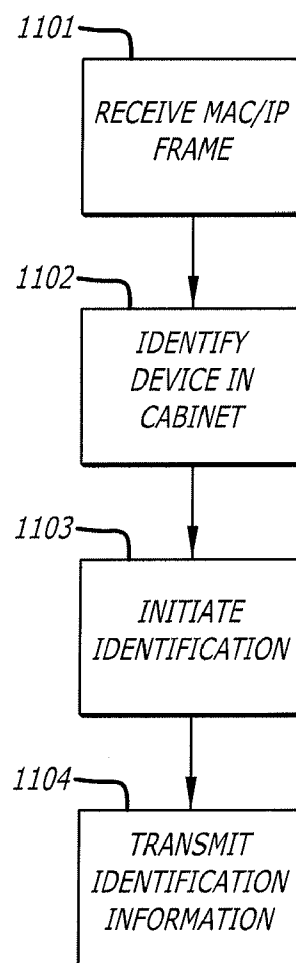


FIG. 11

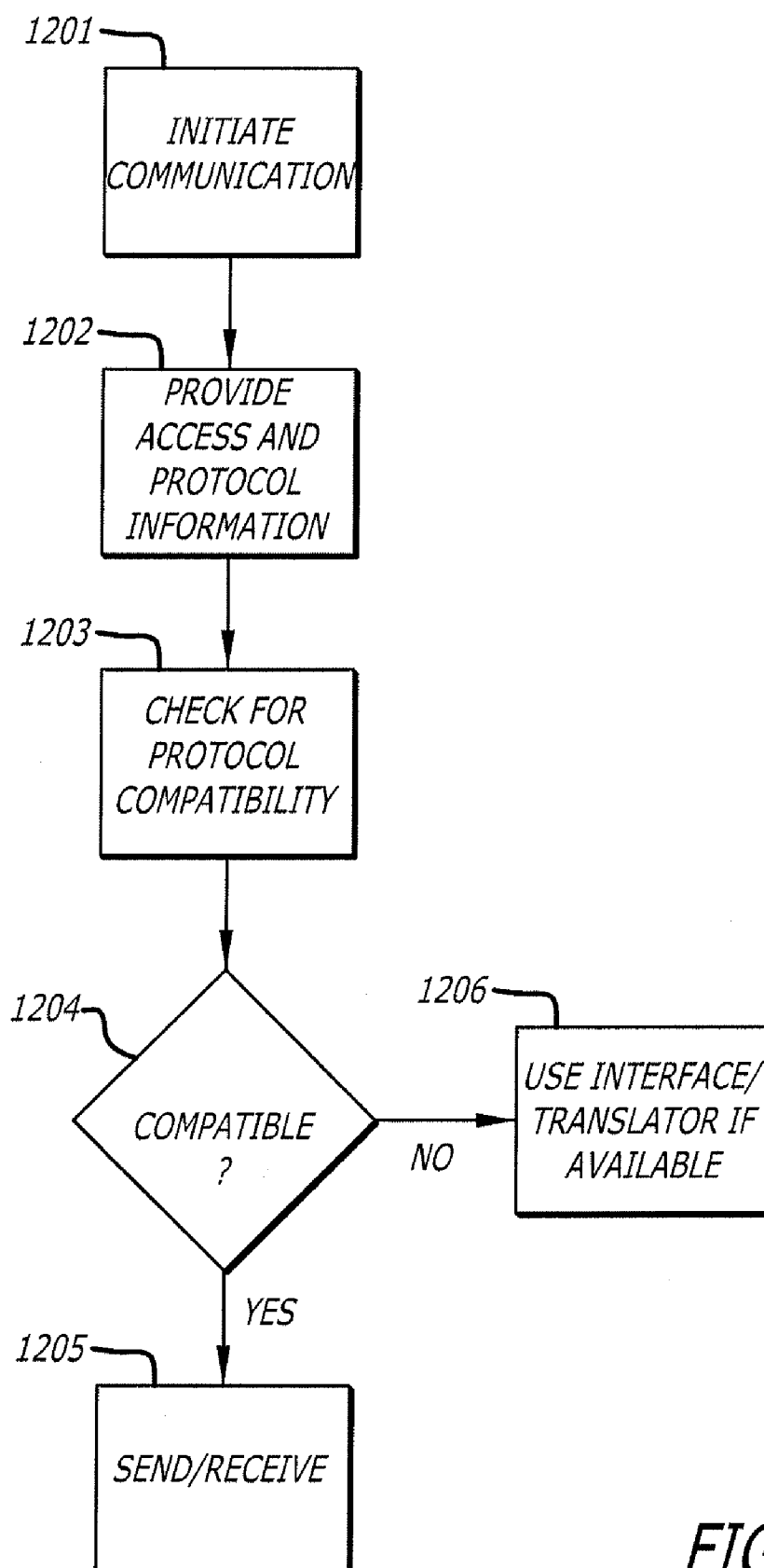


FIG. 12

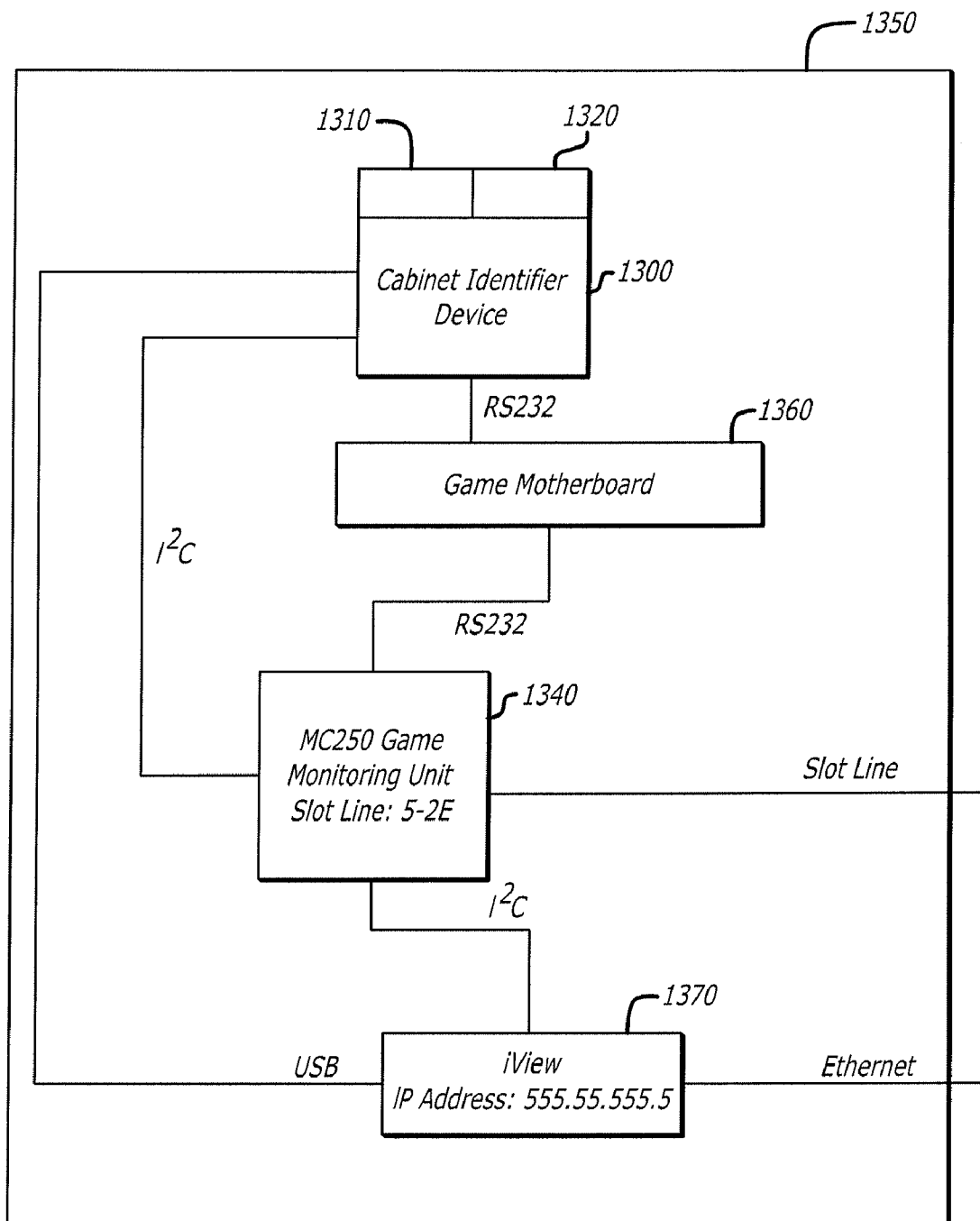


FIG. 13

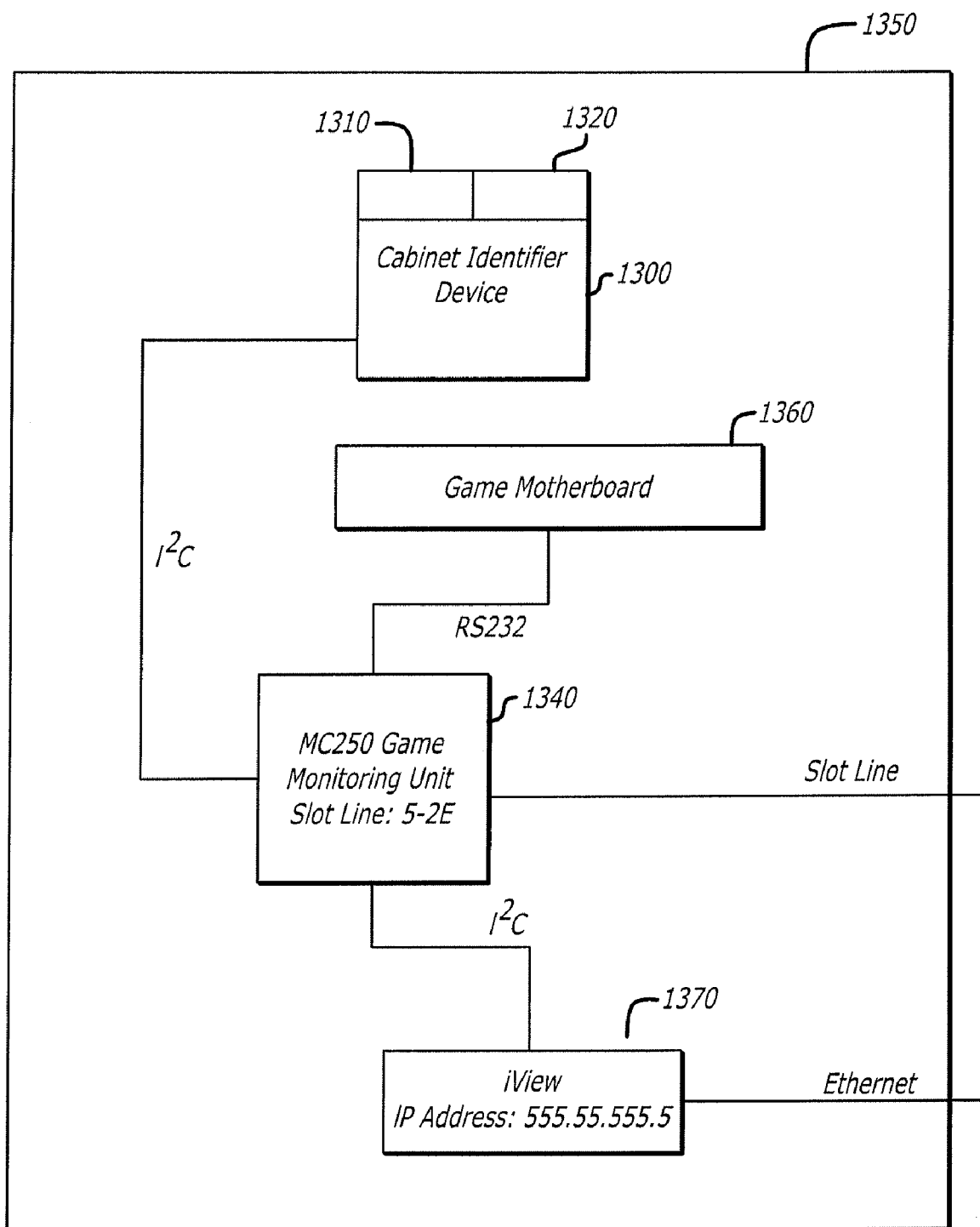


FIG. 14

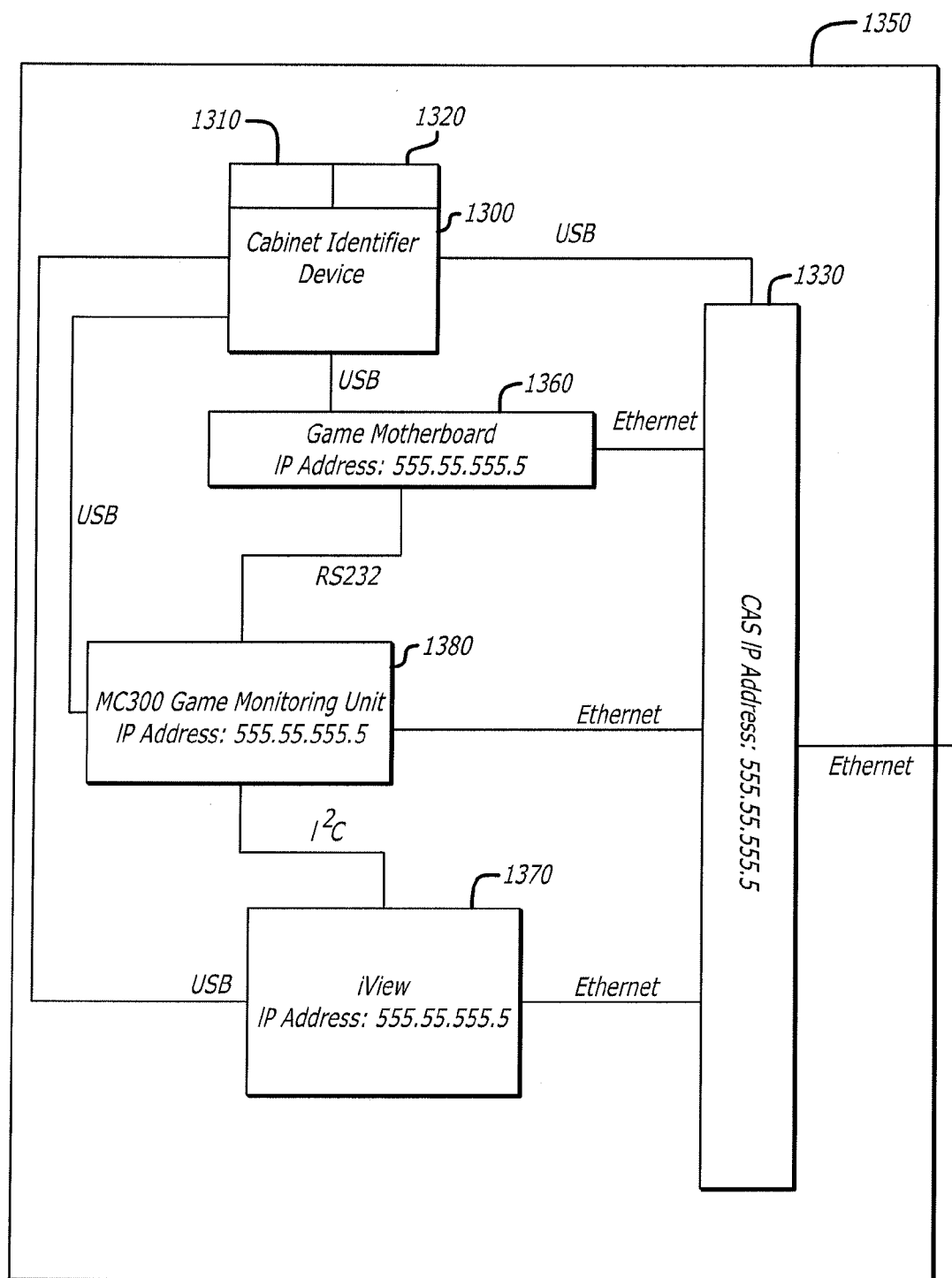


FIG. 15

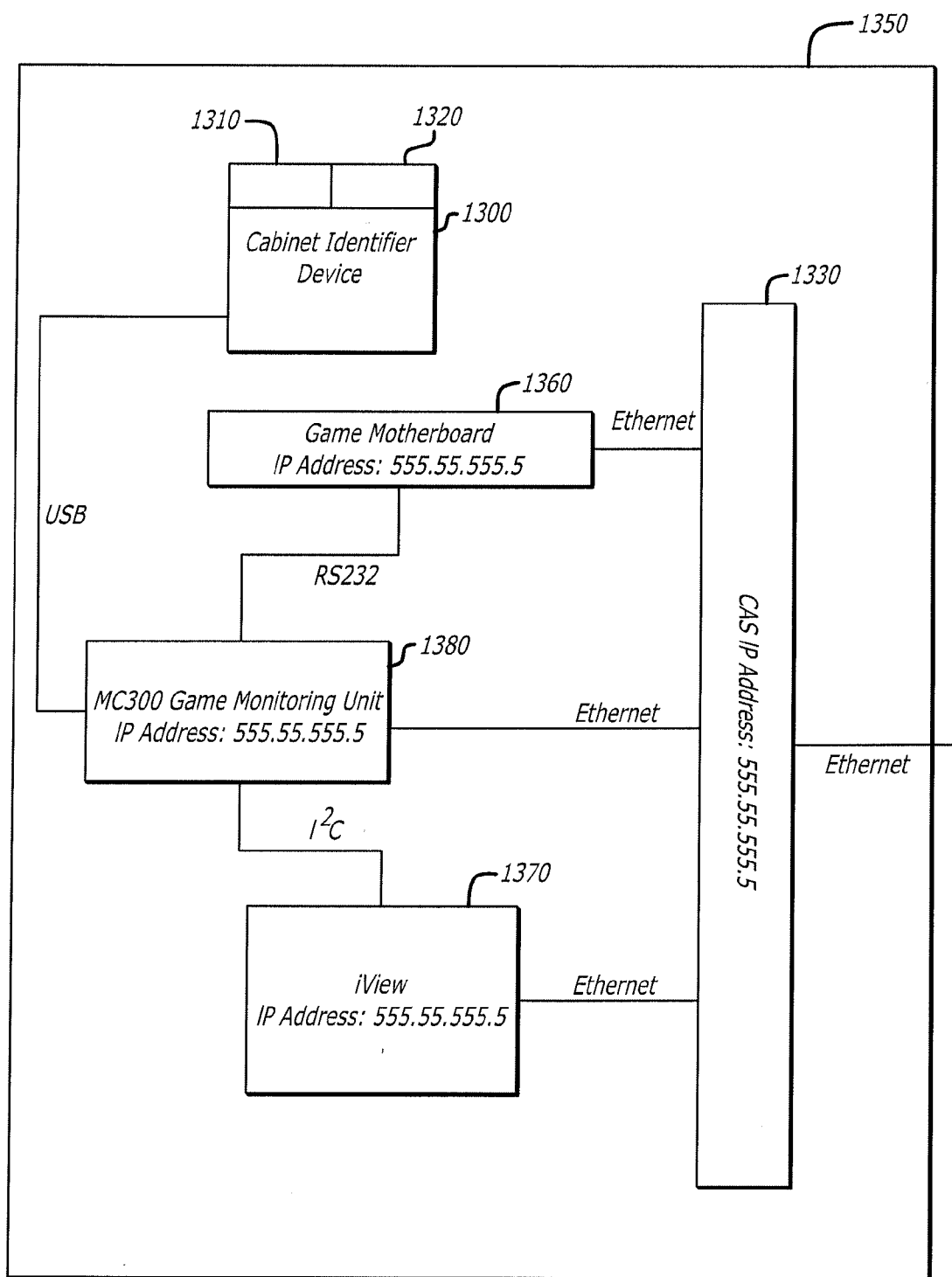
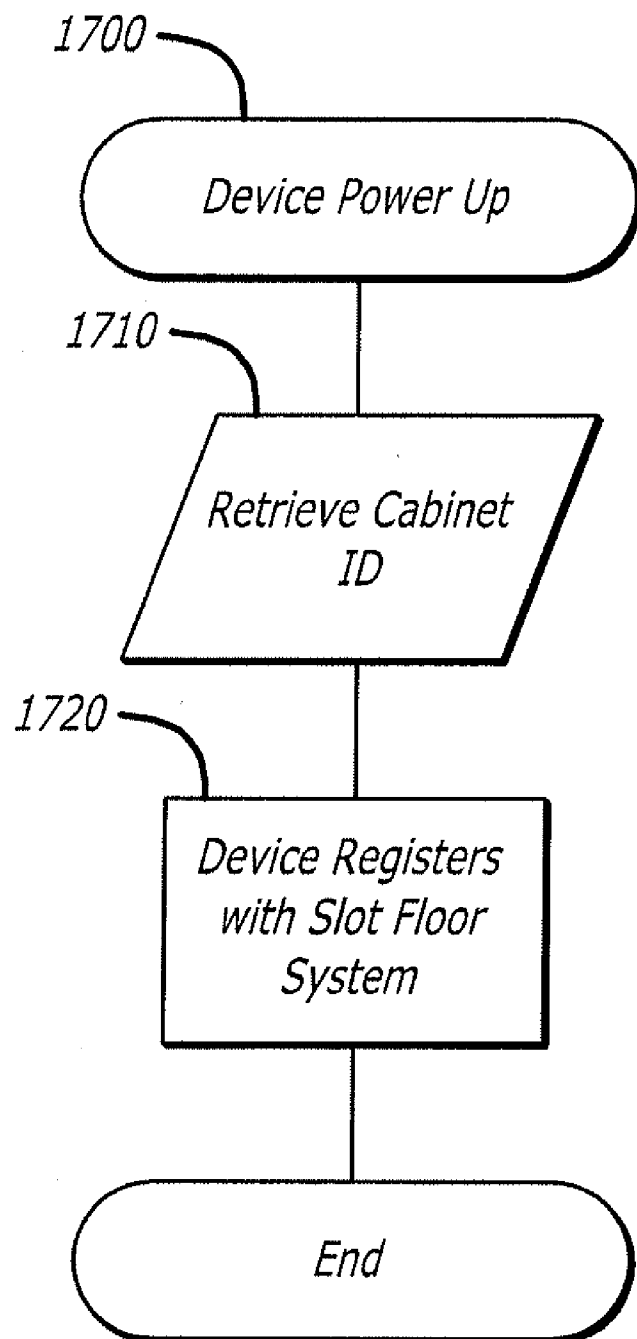


FIG. 16

**FIG. 17**

TRUSTED CABINET IDENTIFICATION METHOD

RELATED APPLICATION

[0001] This patent application is a continuation-in-part of pending U.S. patent application Ser. No. 11/319,034, filed on Dec. 23, 2005 entitled "DEVICE IDENTIFICATION," which is a continuation-in-part of pending U.S. patent application Ser. No. 11/220,781, filed on Sep. 7, 2005 entitled "GAMING NETWORK," all of which are incorporated herein by reference. This patent application is also a continuation-in-part of pending U.S. patent application Ser. No. 11/550,781, filed on Oct. 18, 2006 entitled "Controlled Access Switch," which is incorporated herein by reference.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

FIELD

[0003] The disclosed embodiments relate generally to identification, and more particularly, to a trusted device that provides cabinet identification to components with that cabinet.

BACKGROUND

[0004] In early gaming environments, gaming machines were stand-alone devices. Security of the gaming machines was accomplished via physical locks, security protocols, security personnel, physical and video monitoring, and the need to be physically present at a machine to attempt to breach the security of the gaming machine. By the same token, management of the gaming machines required a great deal of personal physical interaction with each gaming machine. The ability to change parameters of the gaming machine also required physical interaction.

[0005] In view of the increased processing power and availability of computing devices, gaming machines have become customizable via electronic communications and remotely controllable. Manufacturers of gaming equipment have taken advantage of the increased functionality of gaming machines by adding additional devices and features to gaming machines, thereby maintaining a player's attention to the gaming machines for longer periods of time increasing minimum bet and bet frequency and speed of play. This, in turn, leads to the player wagering at the gaming machine for longer periods of time, with more money at a faster pace, thereby increasing owner profits.

[0006] One technique that has been employed to maintain a player's attention at the gaming machine has been to provide players with access to gambling-related information. In this regard, attaching a small electronic display to the gaming device, gambling-related information, as well as news and advertisements can be sent to the player. The gambling-related information may include, for example, information on sports betting and betting options for those sporting events. Additionally, the gambling-related information may also include information such as horse racing and off-track betting. News and advertisements can also maintain a player's

attention by providing the player with access to information ranging from show times, to restaurant and hotel specials, and to world events, thus reducing the need and/or desire of the player to leave the gaming machine.

[0007] In addition, the player may participate in a "premium" promotion where the player is registered with the gaming establishment as a club member when the player inserts an ID card into the gaming machines during play. The player may be rewarded for certain play patterns (e.g. wager amounts, wager totals, payouts, time of play, or the like) and earn redeemable benefits or upgrade of club member status.

[0008] Attempts to distribute gambling-related information and advertisements to players and to allow the recognition of premium membership players have resulted in additional system components that may be attached to the gaming devices. These components for accessing and displaying information for gaming machines may include a keypad, card reader, and display equipment. Other internal components to the gaming cabinet have also been added, such as specialized processors and devices.

[0009] The amount of interactivity and data presentation/collection possible with current processor based gaming machines has led to a desire to connect gaming machines in a gaming network. In addition to the gaming machines themselves, a number of devices associated with a gaming machine or with a group of gaming machines may be part of the network. It has become important for the devices within a gaming machine or cabinet to be aware of each other and to be able to communicate to a control server. Not only is the presence or absence of a network device important, but also the physical location of the device and the ability to associate devices within a particular gaming machine has become a necessary component of a gaming network.

[0010] A gaming network may have a large number of dynamically changing and reconfigurable components. Because of the desire to keep down-time to a minimum, it is important that the population of devices on the network be determinable and verifiable. In the past, this has meant pre-programming knowledge of all other devices into each device, so that communication between devices could take place. Such a requirement of pre-programming or pre-knowledge is too time consuming to be practical in a gaming network environment.

[0011] The current approach to identifying a slot machine relies on slot techs that manually enter the casino's house number (asset number) into the slot machine. The disadvantage of this method is that it is very easy for a slot tech to put in the wrong address (a typo) or to duplicate a slot address. To associate a device in the slot machine a slot technician either manually enters a house number into the device or a device in the slot machine may have a separate device id. This separate device id must be mapped by the host system to the slot machine. Both methods have the disadvantage of putting a wrong device id or duplicating a device id. Finally, the current approach uses a slot technician to manually install public key certificates on a slot machine. This approach has the same limitation as above because if a device is removed for a slot machine, the slot technician must remember to re-associate the device to the slot machine.

[0012] Currently, there is a continuing need to uniquely identify a gaming machine and associate devices within the gaming machine to each other. A unique identifier cannot be guaranteed in a gaming machine, since all components are field replaceable. There is also a need to automate the distri-

bution of public key certificates, which is currently not possible for individual devices with a gaming machine. Current solutions require a human being to setup the initial certificates/public keys.

SUMMARY

[0013] Briefly, and in general terms, the trusted cabinet identification device described herein provides unique cabinet identifiers and security features. More particularly, the trusted cabinet identification device is secured within a cabinet of a gaming machine that is linked to a central server over a network. The gaming machine includes a plurality of components to be identified within the cabinet. The trusted cabinet identification device includes an interface processor and a security device.

[0014] The interface processor communicates unique identifiers and encryption keys to one or more components to be identified within the cabinet. Preferably, the interface processor is capable of communicating in multiple protocols. The security device stores unique identifiers and encryption keys, as well as performs security functions. The trusted cabinet identification device enables components within the gaming cabinet to obtain unique identifiers and encryption keys as needed, and register with the central server over the network.

[0015] Similarly, in another embodiment, a method of identifying components within a gaming machine cabinet that is linked to a central server over a network is provided. As stated above, the gaming machine includes a plurality of components to be identified within the cabinet. The method includes: storing a unique cabinet identifier and encryption key, generating a digital signature from the encryption key, transmitting the unique cabinet identifier to a component to be identified within the cabinet upon power-up of the component, transmitting the digital signature to the component to confirm origin of the unique cabinet identifier, and enabling components within the gaming machine cabinet to register with a game floor manager system on the central server via the network.

[0016] These and other features and advantages of the disclosed embodiments will become apparent from the following detailed description when taken in conjunction with the accompanying drawings, which illustrate, by way of example, the features of the disclosed embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 is a diagram of an embodiment of functional layers of a gaming network;

[0018] FIG. 2 is a block diagram of an embodiment of a gaming network;

[0019] FIG. 3 is a flow diagram of initialization of a network device in an embodiment of a gaming network;

[0020] FIG. 4 is a flow diagram of traffic authentication in an embodiment of a gaming network;

[0021] FIG. 5 is a flow diagram of an attack detection protocol in an embodiment of a gaming network;

[0022] FIG. 6 is a flow diagram illustrating a network device initialization sequence in an embodiment of the gaming network;

[0023] FIG. 7 is a block diagram illustrating examples of possible network attacks;

[0024] FIG. 8 is a block diagram of a gaming machine configuration in an embodiment of the cabinet identification device;

[0025] FIG. 9 is a flow diagram illustrating one embodiment of game machine device management using the cabinet identification device;

[0026] FIG. 10 is a flow diagram illustrating the transmission step of FIG. 9;

[0027] FIG. 11 is a flow diagram illustrating the identification step of FIG. 9; and

[0028] FIG. 12 is a flow diagram illustrating the communication step of FIG. 9.

[0029] FIG. 13 illustrates a gaming machine with a legacy GMU and a trusted cabinet identification device connected to a game motherboard, and an iVIEW connected to the legacy GMU and the trusted cabinet identification device;

[0030] FIG. 14 illustrates a gaming machine with a legacy GMU and a trusted cabinet identification device connected to a game motherboard, and an iVIEW connected to the legacy GMU;

[0031] FIG. 15 illustrates a gaming machine with an IP network-based GMU and a trusted cabinet identification device connected to a game motherboard, and an iVIEW connected to the IP network-based GMU and the trusted cabinet identification device, all of which are connected to a controlled access switch;

[0032] FIG. 16 illustrates a gaming machine with a minimum configuration for the trusted cabinet identification device with a IP network-based GMU, a game motherboard, an iVIEW, and a controlled access switch; and

[0033] FIG. 17 is a logical flow diagram illustrating a device powering up, retrieving a cabinet identifier, and registering with the slot floor system.

DETAILED DESCRIPTION

[0034] The disclosed embodiments are directed to a trusted cabinet identification device that provides a cabinet identifier and other security related information to various components associated with the gaming machine cabinet. The preferred embodiments of the system and method are illustrated and described herein, by way of example only, and not by way of limitation.

[0035] The gaming network described herein proposes an architecture and system that provides an appropriate level of security from network attack. There exist techniques to authenticate and verify individual messages or activities in existing gaming establishment networks relying on proprietary protocols, transport and message formats. However, the gaming network described herein provides additional protection to the network itself particularly when use of commercially based IP equipment is envisioned, above and beyond particular security protocols, for activities and transactions carried on the network. The gaming network is independent of, and in addition to, security techniques for particular transactions or activities.

[0036] Referring now to the drawings, wherein like reference numerals denote like or corresponding parts throughout the drawings and, more particularly to FIGS. 1-7, there is shown one embodiment of the gaming network. As shown in FIG. 1, the network includes a core layer 101 over a distribution layer 102 above an access layer 103. The core layer 101 serves as a gateway between servers and the gaming devices. The core layer 101 is contemplated to be a so-called "back end" layer that resides in an administrative location, separate from the gaming floor, for example, and protected physically and electronically.

[0037] The distribution layer 102 serves to collect traffic between the core layer 101 and the access layer 103. The distribution layer may comprise trunks and switches that route message and signal traffic through the network. The access layer 103 provides a physical interface between the gaming machines (and any of their associated devices) and the rest of the network. This is done via managed switches.

[0038] One embodiment of a network using the layered scheme of FIG. 1 is illustrated in FIG. 2. The core layer 101 includes one or more servers 201 that are coupled via a communication path 202 to one or more switches 203. In one embodiment, the servers and switches of the core layer 101 are located within the gaming establishment premises in a secure administrative area. The servers 201 may, but are not required to be, game servers. The communication path 202 may be hardwire (e.g., copper), fiber, wireless, microwave, Ethernet, wireless Ethernet, or any other suitable communication path that may be protected from attack. In one embodiment, the switches 203 are L2/L3 switches. However, one of ordinary skill in the art will appreciate that other types of switches may be used without departing from the scope or spirit of the disclosed embodiments.

[0039] The distribution layer 102 communicates with the core layer 101 via high bandwidth communications links 204. These links may be copper, fiber, Ethernet, wireless Ethernet, or any other suitable link. If desired, redundant links 205 may be built into the system to provide more failsafe operation. The communications links 204 connect the core layer switches 203 to the distribution layer switches 206. These may be one or more switches, such as L2 switches, for example.

[0040] The distribution layer 102 communicates with the access layer 103 via a high capacity communication link 207. The link 207 may be Ethernet, wireless Ethernet, wire, fiber, wireless, or any other suitable communication link. In the embodiment of FIG. 2, the communication link 207 is coupled to a gaming carousel 208 that comprises a plurality of gaming machines (e.g., 16 gaming machines 215A-215P). A managed switch 209 is coupled to the link 207 to provide an interface switch to a plurality of other managed switches 210 through 213. In the embodiment illustrated, each of the managed switches 210-213 manages four game machines 215. It is understood that the types of switches may be changed without departing from the scope of the disclosed embodiments. Further, switches with more or fewer ports may be substituted and more or fewer tiers of switches in the access layer may be used, as well, without departing from the scope or spirit of the disclosed embodiments. In another embodiment, each game machine has its own managed switch.

[0041] In one embodiment of the gaming network, the network uses TCP/IP sessions between the gaming machines 215 and the servers 201. The TCP/IP sessions are used to exchange private information concerning game operations, game performance, network management, patron information, revised game code, accounting information, configuration and download, and other sensitive information. In one embodiment, sessions may be a single message and acknowledgement, or the sessions may be an extended interactive, multiple transaction session. Other instantiations may include UDP/IP, token ring, MQ, and the like.

[0042] In one embodiment of the gaming network, intrusion detectors provide additional security. In this regard, there may be intrusion detectors located between each layer, such as intrusion detector 220 located between the core layer 101

and the distribution layer 102, and the intrusion detector 221 located between the distribution layer 102 and the access layer 103. In addition, certain sensitive locations or choke points may include intrusion detectors such as the intrusion detector 223 coupled to the switch 209. The intrusion detector 223 may disable the individual ports of switch 209 to isolate attacks while permitting continued operation of the remainder of the gaming network.

[0043] FIG. 8 is a block diagram of an example gaming machine configuration in one embodiment. The gaming machine 215 communicates with the network (e.g. through a managed switch such as switch 210) via communications path 214 which may be Ethernet, wireless Ethernet, wire, fiber, wireless, or any other suitable communication link. The gaming machine 215 may include a communications interface 801 that handles communication between the gaming machine and its associated devices and the remainder of the gaming network. Communication interface 801 is coupled to a game monitoring unit (GMU) 802. The GMU serves as the processor of the gaming machine. An interface referred to as a "SMIB" 803 (smart interface board or slot machine interface board) is coupled to the GMU and to the communication interface 801. SMIB 803 is coupled to one or more peripherals or other devices connected to the gaming machine 215, such as devices 804A to 804N of FIG. 8. In one embodiment, SMIB 803 uses an Ethernet or other high-speed communications link to the communication interface 801, GMU 802, and devices 804A through 804N. In one embodiment, the SMIB includes switching capabilities. In one embodiment, the SMIB is implemented with a Mastercom 300.

[0044] The gaming network may use a number of network services for administration and operation. Dynamic Host Configuration Protocol (DHCP) allows central management and assignment of IP addresses within the gaming network. The dynamic assignment of IP addresses is used in one embodiment instead of statically assigned IP addresses for each network component. A DNS (domain name service) is used to translate between the domain names and the IP addresses of network components and services. DNS servers are well known in the art and are used to resolve the domain names to IP addresses on the Internet.

[0045] Similarly, Network Time Protocol (NTP) is used to synchronize time references within the network components for security and audit activities. It is important to have a consistent and synchronized clock so that the order and the timing of transactions within the gaming network can be known with reliability and certainty. Network information can be gathered centrally at a single workstation by using the Remote Monitoring (RMON) protocol. SNMP (simple network management protocol) allows network management components to remotely manage hosts on the network, thus providing scalability. In one embodiment of the gaming network, SNMPv3 is used to take advantage of embedded security mechanisms to mitigate malicious attacks made against the configuration management function. Still further, TFTP (trivial file transfer protocol) is used by servers to boot or download code to network components.

[0046] In one embodiment, the network may be implemented using the IPv6 protocol designed by the IETF (Internet Engineering Task Force). When using IPv6, the network may take advantage of the Quality of Service (QoS) features available with IPv6. QoS refers to the ability of a network to provide a guaranteed level of service (i.e. transmission rate, loss rate, minimum bandwidth, packet delay, etc). QoS may

be used as an additional security feature in that certain transactions may request a certain QoS as a rule or pursuant to some schedule. Any fraudulent traffic of that nature that does not request the appropriate QoS is considered an attack and appropriate quarantine and counter measures are taken.

[0047] Similarly, the Type of Service (ToS) capabilities of IPv4 may also be used in a similar manner to provide additional security cues for validation of transactions. Again, certain types of transactions may be associated with a particular specific ToS or a rotating schedule of ToS that is known by network monitors.

[0048] Traffic Content

[0049] In an embodiment of the gaming network, the traffic content varies in size and sensitivity. Messages may comprise transactional messages related to game play, such as coin-in. Other messages may be related to management, administration, or sensitive information, such as administrator passwords, new game code, pay tables, win rates, patron personal data, or the like.

[0050] Security

[0051] The gaming network includes network security features, host security features, audit protocols, and design architecture approaches to reduce the likelihood of success of network attacks. Where attacks cannot be prevented, the gaming network attempts to make such attacks expensive in terms of the computational power required, the time, risk, effect, and duration of the attack. Identification of attacks and the rapid recovery from such attacks should be emphasized, as should the limiting of the effect of any attacks.

[0052] Accordingly, the gaming network provides for traffic confidentiality. All nodes within the network exchange information that is confidentially protected. One method for providing confidentially protected data is by using encryption. A number of encryption schemes may be used, such as an FIPS approved encryption algorithm and an NIST specified encryption mode, such as the Advanced Encryption Standard (AES).

[0053] In addition, all nodes within the gaming network apply source authentication and integrity of all traffic. A suitable message authentication mechanism may be, for example, an FIPS approved algorithm such as the Keyed-Hash Message Authentication Code (HMAC) and SHA-1. All nodes automatically drop messages that have been replayed. As noted above, replayed messages are a means of attack on network security.

[0054] Key management mechanisms should be sufficient to resist attack. In one embodiment, a 1024 bit Diffie-Hellman key exchange with a 1024 bit DSA/RSA digital signature is used to render key attacks computationally infeasible. It should be noted that the key sizes are given as examples only. Smaller or greater key size can be used in the gaming network as security recommends. The gaming network should be robust, maintaining the availability of critical services. The network should include protection against misrouting and also discard any traffic that has a source or destination outside of the network. The gaming network should also require a minimum level of authentication and assurance before permitting an additional device on the network and prevent such connection when the assurance is not provided.

[0055] Host protection and security includes secure host initialization where the host performs a self-integrity check upon power-up initialization. All operating system components that are not needed are disabled. When software patches are downloaded to the gaming network, the host verifies

them. The host checks for unused IP ports and disables them prior to connecting to the gaming establishment network. When processing network traffic, any traffic not addressed to the host is dropped from the processing stack as soon as possible. In the gaming network, all service, guest, and default administrator accounts that may be part of the operating system are disabled. In one embodiment, one-time passwords and/or multi-part passwords are used for remote login, if remote login is enabled. The one-time password may itself be a multi-part password. When using a multi-part password, different trusted individuals each hold a part of the multi-part password. The entire password is required for enablement of the system. This prevents any single individual from compromising security. Moreover, all host software components are operated with the lowest privilege necessary for sufficient operation. For example, software that can operate with "user" privilege will do so, to limit its usefulness to an attacker.

[0056] Audit requirements include integrity protection of audit logs from date of creation and throughout their use. Events that are audited in an embodiment of the gaming network include account logon events (both success and failure), account management (both success and failure), directory service events (failure), logon events (success and failure), object access (failure), policy changes (success and failure), privilege use (failure), system events (success and failure), access to a host or networking device logged by user name and the time of access, and all other internal user actions. Anomalous behavior is audited and logged for purposes of evidence for law enforcement and/or attack recognition. Audit information is collected and stored in a secure manner to preserve the chain of evidence. If there is a failure of the audit system, automatic shutdown is initiated.

[0057] The gaming network is designed so that there is no single point of failure that would prevent remaining security features from operating when one is compromised. The gaming network also will continue to operate in the event of bridging to another network, such as the Internet.

[0058] Secure Initialization of Network Devices

[0059] The gaming network provides confidence that a network device is contacting a legitimate DHCP server rather than a spoofed server. The gaming network uses Internet Key Exchange (IKE) in one embodiment. There are a number of modes and phases of IKE. Phase I of IKE includes two modes, referred to as "main mode" and an "aggressive mode". Phase II has a single mode referred to as "quick mode". Main mode takes six packets to complete while aggressive mode takes 3 packets. Quick mode takes 3 packets to complete. In some embodiments, Phase I is used for initialization and Phase II is used to create security for subsequent traffic and messages. FIG. 3 is a flow diagram illustrating the initialization of a network device using main mode of Phase I.

[0060] Phase I is used to authenticate devices to each other and to protect subsequent Phase II negotiations. In the following description, the network device is referred to as the initiator and the server is referred to as the responder. Referring to FIG. 3, at step 301, the initiator sends a first IKE packet to the responder. The packet may or may not include vendor ID's (VID) that can inform the responder of the extensions the initiator supports. Each IKE message includes a mandatory Security Association (SA) that defines how to handle the traffic between the two devices. The SA of the initial packet lists the security properties that the initiator supports, including ciphers, hash algorithms, key lengths, life times and other information. At step 302, the responder replies with an IKE

packet that may or may not include a VID, but does include a mandatory SA payload. At this stage, the packets are not encrypted because there is still no key for encryption.

[0061] The third packet, at step **303**, is from the initiator to the responder and uses the Diffie-Hellman key exchange protocol. The packet contains a key exchange (KE) payload, a NONCE payload, and a certificate request (CR) payload. The public keys are created whenever the phase I negotiation is performed and are destroyed when the phase I SA is destroyed. The NONCE payload is a large random number that has not been used before on the network (“never-used-before”) and is useful in defeating replays. The CR payload includes the name of the Certification Authority for which it would like to receive the responder’s certificate. (Note that the CR can be sent in the third and fourth packets or in first and second packets, as desired).

[0062] At step **304**, the responder returns its own KE, NONCE, and CR in the fourth packet. The third and fourth packets are used by each device to generate a shared secret using public key algorithms. Because only public keys are sent in this exchange, and no encryption key is yet available, the messages are still not encrypted.

[0063] At step **305**, the initiator uses the KE to generate a shared secret and uses it to encrypt the fifth message. The fifth message includes an Identification (ID) payload, zero or more certificate (CERT) payloads (or CRL) and a Signature payload (SIG) that is the digital signature that the responder must verify. The ID payload is used to tell the other party who the sender is and may include an IP address, FQDN (fully qualified domain name), email address, or the like. In an embodiment of the gaming network, it is an IP address. The CERT payload is optional if the initiator or responder cache the public key locally. In an embodiment of the gaming network, the public key is not cached locally and failure to receive a CERT payload is a failure of the negotiation. The SIG payload includes the digital signature computed with the private key of the corresponding public key (sent inside the CERT payload) and provides authentication to the other party.

[0064] At step **306**, the responder sends a message with its ID, CERT, and SIG payloads. When both the initiator and responder have successfully verified the other party’s SIG payload, they are mutually authenticated. The result of the successful negotiation is the Phase I SA.

[0065] After the phase I negotiation is successfully completed, the phase II negotiation can proceed to create SA’s to protect the actual IP traffic with an IPsec protocol. Each of the phase II packets are protected with the phase I SA by encrypting each phase II packet with the key material derived from phase I. Phase II in the gaming network is illustrated in FIG.

4. At step **401**, the initiator sends a message with a number of payloads. The message includes SA and NONCE payloads that are the keying material used to create the new key pair. As noted above, the NONCE payload includes random never-used-before data. The SA payload is the phase II proposal list that includes the ciphers, HMACs, hash algorithms, life times, key lengths, IPsec encapsulation mode, and other security properties. Optionally, the message may include IDi (initiators ID) and IDr (responders ID), which can be used to make local policy decisions.

[0066] At step **402**, the responder replies with a message with the same payload structure as the first message. The initiator replies with a HASH value at step **403**. After phase II is completed, the result is two SA’s. One is used for inbound traffic and the other for outbound traffic.

[0067] Rekeying is done when the lifetime of the SA used for protecting network traffic expires. In one embodiment, PFS (perfect forward secrecy) protocol is used for rekeying. The network ensures the set of secret keys generated by one protocol message exchange is independent of the key sets generated by the other protocol message exchanges. This means compromise of one key set does not lead to compromise of the other sets. Additional protection for network traffic is provided by use of a “virtual private network” (VPN). As a result, all network traffic is protected, and not just TCP/IP traffic.

[0068] In an alternate embodiment, the network may be constrained to a particular regulatory jurisdiction. In this embodiment, a regulatory jurisdiction has its own private key and a multi-tiered approach is used to validate devices. During initialization, a combination key at an administrative location is used to sign messages and data. If there are attempts to communicate outside the jurisdiction, the lack of the regulatory jurisdiction key prevents communication. This is another security feature that is used to limit inside and outside attacks on the gaming network.

[0069] In one embodiment, the system uses a secure key server to store private keys and certificates. The secure key server requires multi-part passwords as described above for access and enablement. The secure key server is resistant to network or Internet attacks, denial of service attacks, and other software or protocol attacks. The secure key server is also resistant to physical attacks such as forced break-in attempts, changes in temperature, changes in pressure, vibration, and attempts to disassemble the secure key server. In one embodiment, any attack attempt results in the destruction of stored keys, certificates, etc, to prevent compromise of the system.

[0070] In another embodiment, a physical transfer of certificates may be implemented as an additional security protection. No game machine or other device may be added to the system without a physical visit and installation of a certificate. In other words, a mere handshaking protocol is not sufficient to add a device onto the system. Rather, a potential new device will require a trusted person or persons to activate the device, install an appropriate certificate, and add it to the network.

[0071] Blocking Illegitimate Traffic

[0072] As described above, the gaming network uses IKE, IPsec, and VPN to protect legitimate traffic from mischief. The gaming network also provides systems to block illegitimate traffic. Firewalls are installed at choke points within the access and distribution layers to isolate network segments from one another. Firewalls can limit the spread of damage from propagating beyond the compromised network segment. The use of NONCE never-used-before random numbers also prevents illegitimate traffic by blocking replay of legitimate messages. IKE and protection of all post-initialization traffic makes it more difficult for illicit messages to achieve successful delivery.

[0073] In addition to detecting false messages using the techniques above, the gaming network reduces the possibility of access to the network by blocking all unused IP ports. Only IP ports required for gaming operation are enabled. To further limit the ability of outside access to the gaming network, private IP addresses are used. Typically IP addresses provide global uniqueness with the intention of participating in the global Internet. However, certain blocks of addresses have been set aside for use in private networks. These blocks of IP

addresses are available to anyone without coordination with IANA or an Internet registry. Since multiple private networks may be using the same block of IP addresses, they lack global uniqueness and are thus not suitable for connection on the global Internet. Private network hosts can communicate with all other hosts inside the private network, both public and private. However, they cannot have IP connectivity to any host outside of the enterprise. Allocation of private network IP addresses may be accomplished pursuant to RFC 1918 protocol.

[0074] In another embodiment, the volume of network traffic is monitored at each link and compared to expected flow rates and/or historical flow rates. Histograms may be generated so that analysis and comparison of flow rates may be accomplished. Heuristic algorithms may be implemented to determine if the flow rate is within an acceptable range. If not, a data leak or attack is assumed and appropriate alarms are triggered. Heavy flow areas can be disabled so that appropriate investigation can be made.

[0075] Detecting and Reacting To Attacks

[0076] Intrusion detection system (IDS) sensors and/or intrusion prevention systems are installed between the core, distribution, and access layers. IDS and intrusion prevention sensors may also be installed at choke points within the access and distribution layers to detect malicious traffic within these layers. One suitable IDS is "arpwatch" (www.securityfocus.com/tools/142) that monitors IP address changes, MAC addresses, flow rate changes, and other network activity and can be configured to notify an administrator when IP/MAC/DID address bindings (e.g. the combination of game machine DID and/or one or more associated device DIDs) change for a device on a gaming network. When a change is detected, automatic isolation procedures may be implemented to isolate the possible intrusion. Subsequent analysis and review by network administrators can determine appropriate responses.

[0077] The system may keep a physical map of the location of the IDS sensors so that when an intrusion is detected, the physical location of the attack can be immediately identified. Security can be dispatched to the location to apprehend the attackers, appropriate systems may be shut down or disabled, and perimeter measures can be taken to increase the chances of securing the attacker.

[0078] FIG. 5 is a flow diagram of one embodiment of the operation of the intrusion detection system of the gaming network. At step 501, the gaming network is initialized and IP addresses are assigned to network devices. This may be accomplished using the technique described in FIGS. 3 and 4 or by any other suitable technique. At step 502, a mapping of the IP addresses of the network devices, their respective MAC addresses, and the DID is performed. This binding should remain stable through a session unless the core layer specifically initiates a change or if a regularly scheduled or anticipated change occurs.

[0079] At step 503, the system monitors the network. Such monitoring may be accomplished by any suitable means for tracking IP/MAC/DID mapping. As noted above, one such method includes Arpwatch. At decision block 504, it is determined if there has been any change to the IP/MAC/DID mapping. If the answer is no, the system continues monitoring the network at step 503. If the answer is yes, meaning that there has been some change in IP/MAC/DID mapping, the system disables the IP address and the network device associated with the MAC address and DID in question at step 505. This step of disabling may also include shutting down ports or

sections of the network to contain or limit any presumed attack on the network. The system notifies the administrator at step 506 so that analysis and correction may begin.

[0080] In an alternate embodiment of the system, the mapping may be between any two of the parameters IP address, MAC, and DID. In addition, there may be multiple devices inside of the gaming machine. In some instances, the DID of the gaming machine may be used exclusively. In other instances, the DID of an associated device such as a reel controller, LED controller, CPU, safeRAM, hard drive, physical cabinet, printer, or other associated devices may be used singly or in combination with the gaming machine DID. Each associated device may have a unique ID (such as a 32 bit hex value) so that the combination of game machine DID and/or one or more associated device DID's results in a unique ID that is difficult to duplicate. Fraudulent communications that lack the requisite binding will be detected easily. Further, malicious hardware that attempts to join the network will lack not only the correct device ID's but also the combination bindings described above.

[0081] In yet another embodiment, the DHCP server is pre-loaded with a list of valid IP addresses, MAC addresses, machine and associated device DIDs, and IP/MAC/DID bindings. If the game machine requesting initialization or permission to join the network is not on the pre-determined list, the machine is not permitted on the network and an attack is logged. An alarm can be triggered so that the attacker can be identified and captured when possible.

[0082] In some instances, it may be useful to use dynamically assigned IP addresses in a gaming network. In such a situation, it is still important to be able to identify with certainty that only valid devices are on the network. In one embodiment, globally unique identifiers (GUIDs) are used to identify managed switches at one or more levels of hierarchy. For example, the switch could be at the game cabinet level, a bank of machine level, and/or a casino level. The GUID is used to positively identify a valid managed switch.

[0083] Associated with each managed switch is what is referred to herein as a "collection" of devices associated with that switch. The DIDs and MAC addresses can be used to identify the devices as being valid members of the collection. The dynamically assigned IP address can then be mapped to the collection so that the members of the network are known, and communication with the collection and its constituent devices can occur. The IP addresses can be subnet IP addresses for members of the collection if desired.

[0084] GUIDs are registered at network creation and when valid devices are added to the system. Once registered, dynamically assigned IP addresses can be properly mapped for communication using the IP address if desired. In another embodiment, each network device has its own GUID that is registered and may be mapped to a dynamically assigned IP address. If desired, the bindings described above may be implemented even with dynamically assigned IP addresses, once the proper mapping has been made using GUIDs.

[0085] Another embodiment takes advantage of GUIDs to create logical collections instead of physical collections. A logical collection may be disparate physically but may be useful for certain management, reporting, or game play operations. By being able to uniquely identify devices and collections, it is possible to create filters that allow communication with subsets of network devices at levels from single devices to collections to all devices and anywhere in between.

[0086] An additional security feature of the gaming network requires a secure boot sequence within each gaming machine and server such that an initial boot is accomplished using code residing in unalterable media. The initial boot code verifies the operating system and all network services it includes. Consequently, network services will not be enabled until the full operating system has been verified as legitimate.

[0087] FIG. 6 is a flow diagram illustrating the boot initialization of a network device, such as a gaming machine in one embodiment of the gaming network. At step 601, the device boots from a locally stored unalterable media. At step 602, the network device establishes security for communication with a network host. This may be accomplished by the IKE phase I method described in FIG. 3. Once secure host communication is established, traffic security is established at step 603. This may be accomplished by IKE phase II, as described in FIG. 4.

[0088] If any of the steps fail in this sequence, communication is terminated and a network administrator is notified. At step 604, the network device submits its operating system for verification. Such verification may be by any desirable method and may be in addition to other network security features. At step 605, the host receives the verification request and checks the operating system of the network device.

[0089] At decision block 606, it is determined if the network device contains a legitimate operating system. If not, the device is disabled at step 607. This process may initiate notice to a network administrator, as well as, disabling of some portion of the network associated with the device in an attempt to mitigate damage from an attack. If the operating system of the network device is legitimate at step 606, the host enables the appropriate network services for the network device at step 608 and operation begins. As noted above, all traffic is protected in the gaming network to some degree. In addition, some traffic includes additional security checks.

[0090] In one embodiment, the game machine provides a secure boot and initial O/S verification as follows. EPROM verification software resides within an input/output processor (IOP). The verification software verifies all EPROMs on the IOP board (i.e., mains and personalities) upon application of power to the game machine. Next, after the application of power to the machine, the BIOS+ performs a self-verification on all of its code. Once satisfactorily completed, the board (e.g. a Pentium class board) begins executing code from the BIOS+ contained in the conventional ROM device. This process verifies the conventional ROM device and detects any substitution of the BIOS+.

[0091] Upon boot-up of the processor, the BIOS+ executes a SHA-1 verification of the entire O/S that is presented. The digital signature is calculated and compared with an encrypted signature stored in a secure location on the game machine using, for example, the RSA private/public key methodology. If the signatures compare, the BIOS+ allows the operating system to boot, followed by the game presentation software. Next, display programs and content are verified, before being loaded into the IOP RAM to be executed for normal game operation.

[0092] During communication, each message is protected using the security of the gaming network. However, certain messages incorporate additional security checks even if the package is considered trustworthy. For example, code downloads may require that they be cryptographically signed and verified before executing. For messages such as these, the digital signature for the code is independent of and in addition

to the authentication provided by VPN and the other network security features. In addition to the digital signature check and verification, the gaming network implements increasing number versioning of network downloaded updates so that rollback attempts may be mitigated or eliminated.

[0093] It may be desired to have some network communication links be wireless instead of hard wired. In such an environment, the gaming network includes wireless intrusion detection mechanisms detecting, for example, 802.1.1a/b/g devices. Such detection has scope beyond network attacks and may detect wireless attacks on the gaming establishment, even if not specifically targeting the gaming network.

[0094] Initialization of Gaming Machine Devices

[0095] One embodiment provides a process for identifying devices coupled to a game machine. This process is described in FIG. 9. At step 901, during initialization, each device (e.g. devices 804A-804N) attempts to communicate with the network and transmits its MAC/IP address. The address is received by a switch in the game machine (e.g. the SMIB 802) and a table of addresses of associated devices is assembled. This table is made available to the devices in the game machine so that the IP addresses of other devices within the gaming machine become available to each device.

[0096] At the identification step 902 each device identifies itself to other devices in the gaming machine. At step 903 a verification process is initiated so that it can be determined if the devices are valid devices on the network. At step 904 devices may begin to transmit data between themselves and to the core layer or other back-end server of the network.

[0097] MAC/IP Transmission

[0098] A description of one embodiment of the MAC/IP transmission of step 901 is illustrated in the flow diagram of FIG. 10. During a boot or initialization sequence 1001, any network-connected device inside the gaming machine will attempt to communicate with the network at step 1002 by sending its MAC/IP address via the SMIB or other switching device. The nature of this initial communication may be for a DHCP or BOOTP configuration, an ARP request, or any other attempt to identify itself to the back-end system. The MAC/IP addresses that are part of these communication attempts are added at step 1003 to a table. This table is managed by the SMIB 803 in one embodiment, or by the GMU 802 in another embodiment. Eventually at step 1004, a table will be generated that contains the MAC/IP addresses of all of the devices in the gaming machine.

[0099] In one embodiment, the devices send only their MAC addresses but the switch or other management device associates an IP address with each MAC address to populate a table. This embodiment may be used when IP addresses are assigned dynamically as described above.

[0100] At step 1005, the switch or GMU, or whichever device is managing the address table, periodically transmits raw Ethernet frames, USB packets, or TCP packets that include a list of the attached MAC/IP addresses associated with that game machine. In one embodiment, the frame is sent on a regular basis (e.g. every three to five seconds) so that other devices can expect that frame and react appropriately if it is not received. The transmitted frame is sent to switches and game machines on the network. In one embodiment, the transmission is via User Datagram Protocol (UDP) but any suitable protocol may be used without departing from the spirit and scope of the disclosed embodiments. In this manner, game machine devices need only be able to recognize the frame to take action. Eventually all of the MAC/IP addresses

of game machine devices are published throughout the network. In this embodiment, there is no necessity of flooding the network with broadcasts frames with address information. This information is distributed organically throughout the network.

[0101] The process in one embodiment is an ongoing process, shown by the return path from step **1005** to step **1002** in FIG. **10**. The tables are rebroadcast periodically by the switch. This rebroadcast allows devices to learn about other new devices that have been added to the network. The rebroadcast also allows a device to know when another device has left the network. At this point in the process the information being collected is pre-authentication. It allows a list of possible devices to be known and addressable so that if the device is valid and authenticated, it can participate on the network.

[0102] Identification

[0103] The identification process **802** is described in conjunction with FIG. **11**. A device receives a MAC/IP transmission frame from the switch at step **1101**. This is an ongoing process during runtime as the switch periodically transmits Ethernet frames containing updated and new MAC/IP address information as described above. At step **1102** the device identifies other devices within the same game machine or cabinet from information in the Ethernet frame. At step **1103** the device initiates an identification communication with one or more other devices in the game machine. The form of this transmission at step **1104** may be as simple as sending an "I'm here" message. In other embodiments, the identification message may include identification information about the device at step **1104**. This information may include information such as the port address, device ID, a preferred communication protocol, and the like. In other embodiments, such information is provided during communication negotiations.

[0104] Verification

[0105] Once two devices have identified themselves to each other, a verification procedure can take place. The verification procedure is intended to establish that the device with which another device is communicating is a valid gaming device. In one embodiment of the disclosed embodiments, verification may be accomplished by using the protocol described herein in connection with FIGS. **3** and **4**. Any suitable verification protocol may be utilized without departing from the scope and spirit of the disclosed embodiments. In-cabinet devices have similar security concerns as other network devices described herein.

[0106] In one embodiment, a verification method such as is described in pending U.S. patent application Ser. No. 10/243, 912, filed on Sep. 13, 2002, and entitled "Device Verification System and Method", assigned to the assignee of the disclosed embodiments, and incorporated by reference herein in its entirety. The disclosed embodiments provide a system and method for verifying a device by verifying the components of that device. The components may comprise, for example, software components, firmware components, hardware components, or structural components of an electronic device. These components include, without limitation, processors, persistent storage media, volatile storage media, random access memories, read-only memories (ROMs), erasable programmable ROMs, data files (which are any collections of data, including executable programs in binary or script form, and the information those programs operate upon), device cabinets (housings) or cathode ray tubes (CRTs). Identifica-

tion numbers or strings of the components are read and then verified. The process of verifying may comprise matching each identification number in a database to determine whether each identification number is valid. In the case where a data file comprises one of a plurality of operating system files, verification of that file, in effect, comprises verifying part of an operating system. For data files, the file names may comprise the identification numbers.

[0107] The database may comprise a relational database, object database, or may be stored in XML format, or in a number of other formats that are commonly known. The database may also comprise an independent system stack of bindings, which comprise numbers, identification strings or signatures in the database for matching or authenticating the components, from manufacturers of the components, each identification number being verified using the binding from the manufacturer of the respective component to verify the component. Especially in the context of smaller devices such as personal digital assistants (PDAs), such a system stack may comprise a subset of one or more global component databases containing bindings from manufacturers of the components, each binding of the subset being associated with at least one of the identification numbers of one of the components in the device.

[0108] Structural components, such as cabinets, may contain an electronic identification chip embedded within them, such as a so-called Dallas chip or an IBUTTON device manufactured by Dallas Semiconductor of Dallas, Tex. These devices allow a unique identifier, placed within a semiconductor or chip, to be placed on a component that may or may not be electronic, such as a computer or gaming machine cabinet. The IBUTTON device is a computer chip enclosed in a 16 mm stainless steel can. The steel button can be mounted, preferably permanently or semi-permanently, on or in the structural component. Two wires may be affixed to the IBUTTON device, one on the top, and one on the bottom, to exchange data between the IBUTTON device and a processor, serial port, universal serial bus (USB) port, or parallel port.

[0109] The matching process may comprise matching each identification number based on the type of component that the identification number identifies. The identification number and the type of component are matched in the database in order to verify that the identification number is valid. Operation of the device may be stopped if any one of the identification numbers is not matched in the database. In the case of a game or gaming machine type of device, a tilt condition message is generated if any one of the identification numbers is not matched in the database.

[0110] The database may consist of a set of signatures, also called bindings. At least with respect to the components that comprise data files or firmware, a well-known hash function, the Secure Hash Function -1, also known as SHA-1, may be used to compute a 160-bit hash value from the data file or firmware contents. This 160-bit hash value, also called an abbreviated bit string, is then processed to create a signature of the game data using an equally well-known, one-way, private signature key technique, the Digital Signature Algorithm (DSA). The DSA uses a private key of a private key/public key pair, and randomly or pseudo-randomly generated integers, to produce a 320-bit signature of the 160-bit hash value of the data file or firmware contents. This signature is stored in the database in addition to the identification number.

[0111] Either contained in the device, or in communication with the device, is a processor and a memory containing executable instructions or a software program file for verification of the components (verification software), which may itself be one of the components to verify. The verification software may be stored on a persistent storage media such as a hard disk device, read only memory (ROM), electrically erasable programmable read-only memory (EEPROM), in the aforementioned CMOS memory, battery-backed random access memory, flash memory or other type of persistent memory. Preferably, the verification software is stored in a basic input/output system (BIOS) on a solid-state persistent memory device or chip. BIOS chips have been used for storing verification software, such as the BIOS+chip used by Bally Gaming Systems, Inc. of Las Vegas, Nev. in their EVO gaming system. Placing the verification software in the BIOS is advantageous because the code in the BIOS is usually the first code executed upon boot or start-up of the device, making it hard to bypass the verification process.

[0112] Alternatively, the verification software may be stored in a firmware hub, which may comprise the part of an electronic device or computer that stores BIOS information. In personal computer hub technology, such as that manufactured by the Intel Corporation of Santa Clara, Calif., a hub is used in place of a peripheral component interconnect (PCI) bus to connect elements of chipsets.

[0113] The persistent storage media may be a removable storage unit such as a CD-ROM reader, a WORM device, a CD-RW device, a floppy disk device, a removable hard disk device, a ZIP disk device, a JAZZ disk device, a DVD device, a removable flash memory device, or a hard card device. However, the database is preferably stored in a non-removable, secure device either within the device being verified, or remotely on a server, in order to enhance security.

[0114] The verification software executes a DSA verification of the data files and firmware components. Also stored in the database is the public key of the private key/public key pair. For each data file and firmware component, as part of the DSA verification, the processor and verification software first computes the hash value of the digital contents of the component using the SHA-1 algorithm. The verification software then processes or authenticates this computed hash value, using the DSA signature verification algorithm, which also takes, as input, the aforementioned public key stored in the database. The verification part of the DSA produces a boolean result (yes or no) as to whether the inputs solve the algorithm. If the algorithm is not solved by the inputs, then an unexpected result is produced, thereby failing to verify the particular component. This may cause a fault tilt to occur to prohibit the loading operation of the device. Otherwise, use of the device is permitted. A detailed description of the DSA can be found in the U.S. government's Federal Information Processing Standards Publication (FIPS) 186-2. That publication describes each step of the DSA signature generation and verification.

[0115] Alternatively, the set of executable instructions may use the Rivest-Shamir-Adleman (RSA) algorithm to verify the components. Using the RSA algorithm, a first abbreviated bit string or hash value is computed from each component's digital contents and encrypted into a digital signature. The digital signature is stored in the database along with the identification number for the component. When the device is verified, the component is verified by computing a second abbreviated bit string computed from the component's digital

contents. The signature is retrieved from the database by searching the database for the identification number. The signature is decrypted to recover the first abbreviated bit string. The component is then verified by comparing the second abbreviated bit string with the first abbreviated bit string. If the first and second abbreviated bit strings do not match, then the component is not verified. As discussed below, this may cause a fault tilt to occur to prohibit the loading operation of the device. Otherwise, use of the device is permitted.

[0116] Instead of creating a digital signature for (i.e., signing) each data file individually, collections of data files may be signed together in order speed up processing. The abbreviated bit strings, hash values, or signatures, also called digests, of the collection of data files are collected into a catalog file, and the catalog is signed as described above.

[0117] Communication

[0118] After verification between devices has been completed, they may begin communication. At step **1201** of FIG. **12**, a device initiates a communication with another device. The sending device may include a section of the first message to provide needed information to the intended recipient. This information may include at step **1202** the type of device, the protocol the device is using, any restrictions related to QOS, and other communication related information. At step **1203** the recipient determines if it can communicate with the sender directly or if an interface is needed at decision block **1204**. If an interface is needed at step **1206**, the sender and receiver may need to communicate through the GMU, for example, if the GMU includes software or firmware for translating appropriately for the devices. If the devices can communicate directly, then messages are sent back and forth using an accepted protocol at step **1205**.

[0119] The disclosed embodiments allow devices to be aware of each other's presence through MAC/IP transmissions. This permits the use of a single network port for each device to use in communicating with each other and with a back-end system. The devices do not need pre-knowledge of the MAC/IP addresses of other devices but can learn them at start up and during run-time. The system also allows a new device to be added to a game cabinet and have it be integrated and identified to the system without extensive IT effort.

[0120] Trusted Cabinet Identification Device

[0121] Referring now to FIGS. **13-17**, a preferred embodiment of the trusted cabinet identification device **1300**, system, and/or method enables the ability to: (1) uniquely identifying a gaming machine on a gaming floor; (2) uniquely associating components in a gaming machine to that gaming machine; and (3) distributing public key certificates.

[0122] As shown in FIG. **13**, in one embodiment, the trusted cabinet identification device **1300** includes two main components, an interface processor **1310** and a security device **1320**. The interface processor **1310** provides communication from external devices through various standard interfaces such as USB, RS-232, and I2C to the security device **1320**. The security device **1320** stores information, such as unique identifiers and encryption keys, as well as providing services such as hashing, encryption, digital signatures and random number generation. Additionally, the security device **1320** may conform to a published standard such as the GEM-PLUS smartcard specification.

[0123] The trusted cabinet identification device **1300** provides a two part solution and substantial benefits in terms of providing access to security and identification information by

external devices that would not ordinarily be able to communicate with a dedicated single interface security device. Using an off the shelf microcontroller as a front end for this communication service, the ability to support nearly any external device becomes the simple matter of a firmware upgrade.

[0124] Notably, the trusted cabinet identification device **1300** has the ability to communicate with multiple devices within a gaming machine. Without the use of the trusted cabinet identification device **1300**, prior techniques have required a unique ID associated with a single device. Multiple ID chips were installed in a gaming cabinet, with each device in the gaming cabinet wired to one chip. The host system then had to have a master list of all ID chips and the associated gaming machine.

[0125] In one embodiment, the trusted cabinet identification device **1300** and the controlled access switch **1330** (shown in FIGS. **15** and **16**) complement one another in identifying cabinet devices. The controlled access switch **1330** groups devices that are based on an IP network. A preferred embodiment of the trusted cabinet identification device **1300** communicates with multiple protocols including RS232, USB, I²C, and Ethernet. This enables the trusted cabinet identification device **1300** to provide a fixed, tamper-resistant asset identifier in conjunction with the controlled access switch **1330** grouping function, as well as supporting cabinet identification with legacy products such as the legacy GMU **1340** (game monitoring unit, i.e., the GMU model MC250).

[0126] Since the controlled access switch **1330** is not permanently attached to the cabinet of the gaming machine **1350** and is designed to be field replaceable in case of a malfunction, a controlled access switch **1330** may not be permanently identified as associated with a particular asset number without an asset identifier like the trusted cabinet identification device **1300**. If the controlled access switch **1330** device is removed from the gaming machine **1350**, the slot floor system has to be updated to reflect the new controlled access switch **1330** device in the gaming machine **1350**. The trusted cabinet identification device **1300** is designed to be permanently attached in a tamper evident manner to the gaming machine cabinet. By permanently attaching the trusted cabinet identification device **1300** to the cabinet, the slot floor system does not have to be manually updated when any component is replaced, including the controlled access switch **1330**.

[0127] As shown in FIG. **13**, a gaming machine **1350** with a legacy GMU **1340** is attached to a gaming machine's game motherboard **1360**. An iView **1370** (a system gaming and/or player tracking user interface and processor, or other comparable component) is also attached to the GMU **1340**. Without the trusted cabinet identification device **1300**, financial transactions through the iView **1370** are problematic. Since the iView **1370** is not directly connected to the game motherboard **1360**, the iView **1370** is reliant upon a slot technician to manually configure the game motherboard **1360**, the GMU **1340** and the iView **1370** with the correct slot asset number. If any of these values are incorrect, the financial transactions are credited to the incorrect device. With the trusted cabinet identification device **1300**, even if a slot technician incorrectly configures any devices in the gaming machine **1350**, the slot floor system can identify the issue for resolution. However, the slot floor system can still function properly and allow financial transactions through any of those devices without disabling the gaming machine **1350**. Listed below is the table for the slot floor system that would be used to associate the

devices in a gaming machine. Since the GMU **1340**, iView **1370**, and the game motherboard **1360** have the same cabinet identifier, then all of these devices are in the same cabinet.

DEVICE	CABINET IDENTIFIER	NETWORK ADDRESS
GMU 1340 (MC250)	425F3FD4-62AB-33FS-BC2E-25271A4E5523	5-2E
iView 1370	425F3FD4-62AB-33FS-BC2E-25271A4E5523	172.26.247.3
Game Motherboard 1360	425F3FD4-62AB-33FS-BC2E-25271A4E5523	

[0128] In FIG. **14**, the minimum configuration required for an embodiment of the trusted cabinet identification device **1300** configured for use with the legacy GMU **1340** (MC250) is shown. The GMU **1340** is connected to the trusted cabinet identification device **1300**, which is then attached to the rest of the devices in the gaming machine **1350**. In this configuration, the GMU **1340** retrieves the cabinet identifier and then passes this information to the game motherboard **1360** and iView **1370**. This configuration of the trusted cabinet identification device **1300** has the benefit of uniquely identifying the gaming machine cabinet for each device.

[0129] As shown in FIG. **15**, the gaming machine **1350** has had the GMU **1340** (model MC250) replaced with an IP network-based GMU **1380** (model MC300). The controlled access switch **1330** in this embodiment can associate the game motherboard **1360**, the GMU **1380**, and the iView **1370** to being inter-related. If the controlled access switch **1330** uses the cabinet identifier from the trusted cabinet identification device **1300**, then in a situation where the controlled access switch **1330** fails, the association of the device will not have to be updated on the slot floor system. However, if the game motherboard **1360** was not connected to the controlled access switch **1330**, then the controlled access switch would not be able to associate the game motherboard to the other devices in the gaming machine cabinet.

DEVICE	CABINET IDENTIFIER	NETWORK ADDRESS
IP Network based GMU 1380 (MC300)	3F2504E0-4F89-11D3-9A0C-0305E82C3301	5-2E
iView 1370	3F2504E0-4F89-11D3-9A0C-0305E82C3301	172.26.247.3
Game Motherboard 1360	3F2504E0-4F89-11D3-9A0C-0305E82C3301	
Trusted Cabinet Identification Device 1300	3F2504E0-4F89-11D3-9A0C-0305E82C3301	

[0130] In FIG. **16**, the minimum configuration for the trusted cabinet identification device **1300** with a GMU **1380** (MC300) is shown. In this embodiment of the trusted cabinet identification device **1300**, the GMU **1380** retrieves the cabinet identifier from the trusted cabinet identification device **1300** and then relays the cabinet identifier to all of the other attached devices.

[0131] In a preferred embodiment, the trusted cabinet identification device **1300** permanently attaches to the gaming machine cabinet and enables multiple devices in a gaming machine **1350** to query the device for the trusted cabinet

identification device's unique identifier. The use of the trusted cabinet identification device **1300** guarantees globally unique identifiers.

[0132] In one embodiment, the trusted cabinet identification device **1300** provides a globally unique identifier, is powered from USB or EPI ports, and is accessed only on power-ups of attached devices to minimize power consumption and processing requirements. Additionally, the trusted cabinet identification device **1300** has the following interconnects: (1) USB (enabling multiple hosts to communicate with the trusted cabinet identification device)—as used in GSA devices and controlled access switch type network switches; (2) EPI (I²C)—as used in some manufacture-specific GMUs; (c) Serial RS232—as used in legacy gaming manufacturers and some manufacture-specific NT boards.

[0133] Further, in some embodiments the trusted cabinet identification device **1300** is epoxied (or otherwise secured) into gaming machine cabinet for tamper evident security, since the gaming machines **1350** already have physical security for the cabinet. If the component devices (e.g., gaming machine motherboard **1360**, NVRAM, hard drive, GMU **1340** or **1380**, iVIEW **1370**, controlled access switch **1330**, or the like) are replaced in the field, the slot floor system can easily recreate the association. This is accomplished by having the attached devices send the trusted cabinet identification device **1300** a unique identifier to the slot floor system upon registration with the system.

[0134] In one embodiment, the trusted cabinet identification device **1300** has the capability to sign a message from an attached device. This signature uses a shared key with slot floor system to guarantee that the message did indeed originate from the gaming machine **1350**. The trusted cabinet identification device **1300** is a trusted device in the gaming machine **1350** that other attached devices use to build a trusted connection to the slot floor system.

[0135] Referring now to FIG. 17, a logical flow diagram is shown that presents a component device powering up, at step **1700**, retrieving a cabinet identifier, at step **1710**, and registering with the slot floor system, at step **1720**. More specifically, each of the component devices that are connected to the trusted cabinet identification device **1300** retrieve the cabinet identifier from the device **1300** on component device power up. The cabinet identifier is not intended to be stored on each component device. In retrieving the cabinet identifier, the trusted cabinet identification device supports USB, I2C, Ethernet, and RS232 connections. Next, the component devices register with the slot floor system. Each component device registers with the slot floor system in a device specific way with the appropriate network identifier and the cabinet identifier. The network identifier can be an IP address or a slot line address. The slot floor system can then associate all devices in the gaming machine with the cabinet in which they reside.

[0136] Although the disclosed embodiments have been described in connection with in-cabinet devices identifying themselves to each other, it is not limited to such an application. The disclosed embodiments may be used to provide identification of any network devices by organically updating identification information periodically in Ethernet frames. In addition, the disclosed embodiments are not limited to the specific network configuration described herein. Rather, the system can work with any number of network configurations without departing from the scope and spirit of the disclosed embodiments.

[0137] It will be apparent from the foregoing that, while particular forms of the disclosed embodiments have been illustrated and described, various modifications can be made without departing from the spirit and scope of the disclosed embodiments. Accordingly, it is not intended that the disclosed embodiments be limited, except as by the appended claims.

What is claimed is:

1. A method of identifying components within a gaming machine cabinet that is linked to a central server over a network, wherein the gaming machine includes a plurality of components to be identified within the cabinet, the method comprising:

- storing a unique cabinet identifier and encryption key;
- generating a digital signature from the encryption key;
- transmitting the unique cabinet identifier to a component to be identified within the cabinet upon power-up of the component;
- transmitting the digital signature to the component to confirm origin of the unique cabinet identifier;
- enabling components within the gaming machine cabinet to register with a game floor manager system on the central server via the network.

2. The method of claim 1, wherein the transmitting of the unique cabinet identifier to the component to be identified within the cabinet is transmittable is multiple protocols including USB, RS232, I2C, and Ethernet.

3. The method of claim 1, further comprising performing security functions includes hashing, encryption, digital signature generation, random number generation, or combinations thereof.

4. The method of claim 1, further comprising performing the transmitting of the unique cabinet identifier using a trusted cabinet identification device that is secured within the cabinet in a tamper-resistant manner.

5. The method of claim 1, further comprising enabling a message to be digitally signed regarding a unique identifier to confirm message origination from the gaming machine.

6. The method of claim 5, wherein the digitally signed message is sent to a game floor manager system on the central server to create a trusted connection.

7. The method of claim 1, further comprising performing the transmitting of the unique cabinet identifier using a trusted cabinet identification device, wherein the trusted cabinet identification device is only directly connected to a game monitoring unit.

8. The method of claim 7, further comprising performing the transmitting of the unique cabinet identifier using a trusted cabinet identification device, wherein the trusted cabinet identification device is indirectly connected to other components to be identified within the cabinet via the game monitoring unit.

9. The method of claim 1, further comprising performing the transmitting of the unique cabinet identifier using a trusted cabinet identification device, wherein the trusted cabinet identification device is directly connected to all components to be identified within the cabinet.

10. The method of claim 1, further comprising performing the transmitting of the unique cabinet identifier using a trusted cabinet identification device, wherein the trusted cabinet identification device is connected to a controlled access switch that assists the components to be identified within the cabinet with transmission of their corresponding addresses.

11. A method of identifying components within a gaming machine cabinet that is linked to a central server over a network, wherein the gaming machine includes a plurality of components to be identified within the cabinet, the method comprising:

storing unique identifiers and encryption keys in a security device, wherein the security device performs security functions;

generating a digital signature from the encryption key;

transmitting the unique cabinet identifier to a component to be identified within the cabinet upon power-up of the component using an interface processor, wherein the interface processor is capable of communicating in multiple protocols;

transmitting the digital signature to the component to confirm origin of the unique cabinet identifier; and

enabling components within the gaming machine cabinet to register with a game floor manager system on the central server via the network.

12. The method of system **11**, wherein the multiple protocols include USB, RS232, I2C, and Ethernet.

13. The method of system **11**, wherein the security functions includes hashing, encryption, digital signature generation, random number generation, or combinations thereof.

14. The method of system **11**, wherein the trusted cabinet identification device is secured within the cabinet in a tamper-resistant manner.

15. The method of system **11**, wherein the trusted cabinet identification device enables a message to be digitally signed regarding a unique identifier to confirm message origination from the gaming machine.

16. The method of system **15**, wherein the digitally signed message is sent to a slot floor system on the central server to create a trusted connection.

17. The method of system **11**, wherein the trusted cabinet identification device is only directly connected to a game monitoring unit.

18. The method of system **17**, wherein the trusted cabinet identification device is indirectly connected to other components to be identified within the cabinet via the game monitoring unit.

19. The method of system **11**, wherein the trusted cabinet identification device is directly connected to all components to be identified within the cabinet.

20. The method of system **11**, wherein the trusted cabinet identification device is connected to a controlled access switch that assists the components to be identified within the cabinet with transmission of their corresponding addresses.

* * * * *