GAMING STANDARDS ASSOCIATION™

Software Verification and Authentication in a Gaming Device

White Paper

Date: February 15, 2000

39355 California Street, Suite 307 • Fremont, CA 94538 Tel: 510.744.4007 • Fax: 510.608.5917 • www.gamingstandards.com

DRAFT

39355 California Street, Suite 307 • Fremont, CA 94538
Tel: 510.744.4007 • Fax: 510.608.5917 • www.gamingstandards.com

1 Introduction

Gaming machine development has progressed further in the last several years than in all of the years since their invention. This is especially true of video based gaming machines. No longer are poker hands simply being dealt or are keno cards checked off as random numbers are drawn. Bonus games are added and have increasingly sophisticated animation. Video monitors have higher screen resolution and color depth. And audio has gone from a mono speaker to stereo surround sound. All of this requires increasing amounts of program media storage to store the executable code, graphic images and sound data. Gaming machine manufacturers have had to turn to other memory storage devices not typically associated with gaming machines. This change in technology has outgrown the current methods of ensuring the integrity of the gaming machine's software. The scope of this paper is to discuss the verification and authentication methods associated with this change in technology.

Strictly speaking, verification and authentication are two separate processes. Verification is the process of determining that the program code or other stored data is intact and is not corrupted by storage media failures. In contrast, authentication is the process of determining that the program code or other stored data are authentic copies of the approved components.

2 Verification

In order to check for corruption in the storage media, the gaming machine software performs the verification process that is internal to the gaming machine. At selected events such as power resets, door closures, and before game play, the software performs a mathematical calculation and uses the software itself that resides in the storage media as the data for this calculation. With a properly chosen mathematical algorithm, errors in the data, or in this case, the storage media, can be detected.

For EPROM based storage media, the gaming regulations generally do not specify the exact algorithm that must be used, only that it must be able to detect 99.99 percent of all possible media failures[1]. The most common algorithm is the cyclic redundancy check (CRC). The CRC of the software data is calculated and the results are stored in the EPROM itself at the time it is programmed. During the operation of the game, the software calculates the CRC of itself and compares it with the results previously stored in the EPROM. If the results match, it is assumed that the software is intact and therefore, the program storage media is free from any failures.

3 Authentication

In order to authenticate the software in the storage media, the contents of the media are checked against a known good copy.

For EPROM based storage media, there are two common methods of authentication. Both involve a process that is external to the gaming machine. The EPROM device(s) is(are) physically removed from the machine. In the first method, the EPROM is inserted into an electronic device and the data is read. The data is then compared, bit by bit, with the data of a master EPROM that has been kept in a secure location. If the results of the comparison indicate that both EPROM's have the same binary image, that is, every single bit in every single location matches exactly, then the software is considered to be authenticated.

---

[1] Example, Nevada Regulation 14, Technical Standards for Gaming Devices, Standard 1, Integrity of Devices, Section 1.080 Control Program Requirements.

In the second method, the EPROM is inserted into an electronic device. However, instead of simply reading the contents, an algorithm is performed on the data, in much the same way as during verification. The result of the algorithm is referred to as the EPROM's signature. This signature is compared with the previously calculated signature of the master EPROM. If the calculated signature matches that of the master EPROM signature, then the software is considered to be authenticated.

The methods for verification and authentication discussed so far work reasonably well for EPROM technology as the storage media. However, the methods do not work well or simply do not work at all for the other storage technologies gaining use. Case in point is high capacity storage devices such as hard drives.

4 High Capacity Storage Devices

These storage devices typically have from 10 to even 40,000 times the storage capacity of a single EPROM chip. Thus the attractiveness in using these devices to store the increasing amounts of graphics and sound data in gaming machines. There are several devices in use today and can be divided into two categories: rotating media and solid state.

- Hard Drives
- CD-ROM Drives
- DVD Drives
- Flash

Unlike EPROM technology, these are mass storage devices only. The program does not actually execute directly from them. The program is copied into higher speed memory and executed from there. Although flash could be used in a system where the program can execute directly from the flash device, there are other considerations of why it generally does not. That consideration is the operating system. The operating system is the software that controls the resources of the electronic hardware. There are many variations of operating systems to select from. They range from proprietary to those commercially available from third parties. They all have one thing in common though, a format to store that data. The format is how the data is organized in the device so that the operating system and application can access it.

In addition to the software considerations of using these devices, there are hardware considerations as well. The first is the many interface technologies available. The interface dictates how to connect the devices, both electrically and mechanically, to the computer. Below is a summary of the various software data formats and hardware interfaces for these mass storage devices.

	SOFTWARE	HARDWARE	
DEVICE	Data Format	Electrical	Mechanical
Hard Drive	FAT16 FAT32 Unix	IDE/EIDE ATAPI SCSI/SCSI 2	40-pin connector 58-pin connector
CD-ROM Drive	ISO9660 Unix	IDE/EIDE ATAPI SCSI/SCSI 2	40-pin connector 58-pin connector

DEVICE	SOFTWARE	HARDWARE	
	Data Format	Electrical	Mechanical
DVD Drive		IDE/EIDE ATAPI	40-pin connector 58-pin connector
Flash	Linear FAT16 FAT32	IDE ATAPI PCMCIA	IC – chips PCMCIA-Type I PCMCIA-Type II PCMCIA-Type III CompactFlash

5 Levels of Verification and Authentication

Many manufacturers have physical constraints in choosing a location to mount the storage device such as cooling requirements, size, maximum cable length, etc. This sometimes results in a less than optimal location for easy removal. It would be beneficial for the manufacturers to have all verification and authentication take place without removing the storage device. However, there may be instances where the storage device would need to be removed.

Therefore different levels of verification and authentication can be defined based on length of time to verify or authenticate as well as level of security such as routine or resolving a dispute.

Define two levels of verification:

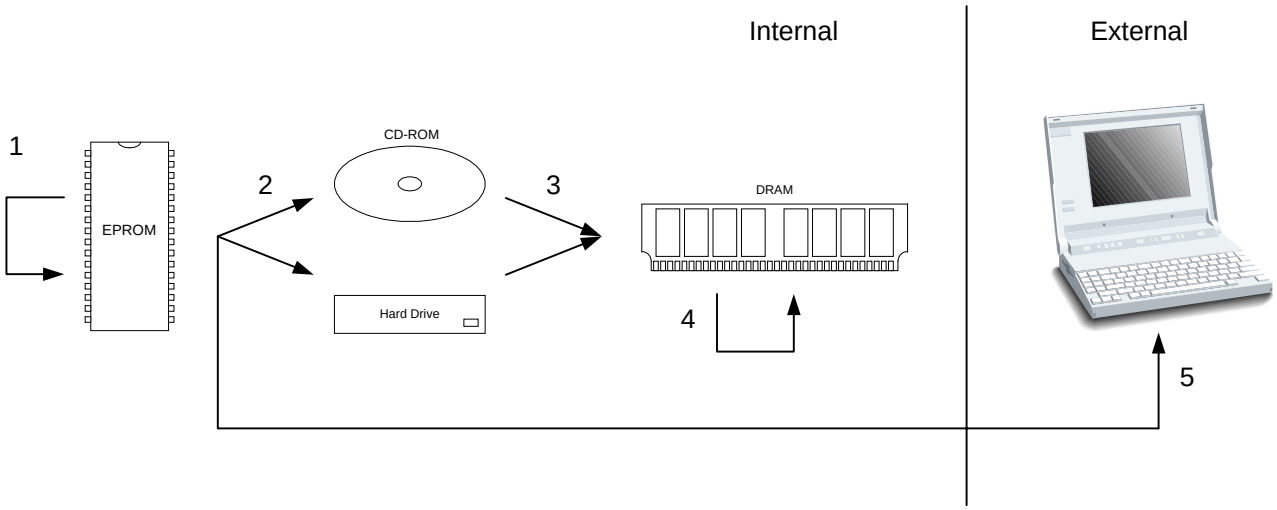
Level	Process	Data
0	Internal	Control Program
1	Internal	Control Program, Graphic data, Sound data

Define four levels of authentication:

Level	Process	Data
0	Internal	Control Program
1	Internal	Control Program, Graphic data, Sound data
2	External	Control Program
3	External	Control Program, Graphic data, Sound data

Process: Internal Storage media remains internal to the gaming machine
External Storage media is removed from the gaming machine

6 Verification/Authentication Environment



Verification takes place over steps 1, 2, 3, and 4, in the above figure.

In Step **1**: Verify nonvolatile storage media

In Step **2**: Verify storage media is intact and not corrupted by media failure.

In Step **3**: Verify code as it is brought into DRAM to execute is intact and not corrupted by media failure.

In Step **4**: "Continuously" verify code/data in DRAM is intact and not corrupted by media failures.

Authentication takes place over steps 1, 2, and 4, in the above figure.

In Step 1: Authenticate code/data in storage media itself

In Step 2: Authenticate code/data as it is brought into DRAM to execute

In Step 4: Authenticate code/data in storage media itself and send results via serial port to laptop

The internal authentication involves performing a given authentication algorithm and sending the results to a laptop or some other data collection device. A standard format for sending the results is beneficial to the manufacturers and regulators alike. The format should have a minimum of information to make its parsing easier. The essential information includes the following:

- Gaming machine manufacturer ID
- Authentication file size
- Authentication level
- Authentication method used
- File(s) verified and their Authentication data (digest)
- Authentication file signature.

This allows a lot of flexibility for the manufacturers to select their own authentication algorithms.