US Patent & Trademark Office
Patent Public Search | Text View

United States Patent	12386712
Kind Code	B2
Date of Patent	August 12, 2025
Inventor(s)	Ishibashi; Kota

Storage system

Abstract

A storage system is protected from tampering of software executed by the storage system. The storage system includes a first storage controller and a second storage controller. The first storage controller includes a first input and output controller configured to input and output host data, and a first management controller. The second storage controller includes a second input and output controller configured to input and output host data, and a second management controller. The first management controller is configured to store a backup of software of at least one of the second storage controller or the first input and output controller. A copy of tampered software of the at least one is stored. The tampered software of the at least one is recovered by the backup.

Inventors:	**Ishibashi; Kota (Tokyo, JP)**
Applicant:	**Hitachi, Ltd.** (Tokyo, JP)
Family ID:	**1000008752028**
Assignee:	**Hitachi Vantara, Ltd. (Yokohama, JP)**
Appl. No.:	**18/472085**
Filed:	**September 21, 2023**

Prior Publication Data

Document Identifier	Publication Date
US 20240256401 A1	Aug. 01, 2024

Foreign Application Priority Data

JP	2023-012912	Jan. 31, 2023

Publication Classification

Int. Cl.:   **G06F11/07** (20060101); **G06F11/14** (20060101)

U.S. Cl.:

CPC   **G06F11/1469** (20130101); G06F2201/865 (20130101)

Field of Classification Search

CPC: G06F (11/1469); G06F (2201/865)

USPC: 714/15

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
8745001	12/2013	Goldschmidt	707/645	G06F 11/1448
12066913	12/2023	Fukushima	N/A	G06F 11/0727
2005/0010617	12/2004	Chen	N/A	G06F 11/1417
2021/0224387	12/2020	Uragaki	N/A	H04N 1/00074
2022/0382533	12/2021	Kajimoto	N/A	G06F 11/1417

FOREIGN PATENT DOCUMENTS

Patent No.	Application Date	Country	CPC
2012-022721	12/2011	JP	N/A

Primary Examiner: Ko; Chae M

Attorney, Agent or Firm: Foley & Lardner LLP

Background/Summary

CLAIM OF PRIORITY
(1) The present application claims priority from Japanese patent application JP 2023-012912 filed on Jan. 31, 2023, the content of which is hereby incorporated by reference into this application.
BACKGROUND OF THE INVENTION
1. Field of the Invention
(2) The present invention relates to a storage system, and more specifically to inspection of software in the storage system.
2. Description of Related Art
(3) Data storage is a basic function of a computer system. In many computer systems, when a large amount of data is handled, the data is stored in a storage apparatus. The storage system stores the data in a storage medium (storage drive) such as a hard disk drive (HDD) or a solid state drive (SSD), and writes and reads the data according to an instruction from an outside.
(4) In recent years, it is important to conform to supply chain security, and a firmware tampering prevention function is implemented in servers. In addition, JP2012-022721A discloses a virus checker that checks virus infection for a file, and a technique of moving a file in which the virus infection is detected by the virus checker to a non-executable storage area in a storage apparatus (see, for example, Abstract).

CITATION LIST
Patent Literature
(5) PTL 1: JP2012-022721A
SUMMARY OF THE INVENTION
(6) A storage system that is required to continuously work at all times has a lower tolerance for reactivation of the system even when a fault occurs as compared with a server. This also applies to ensuring of a security level. It is required that the storage system operates while ensuring the security level in addition to I/O performance in related art. In addition, in the storage system, it is necessary to perform detailed cause analysis on and take countermeasures against tampering of firmware.

(7) A storage system according to an aspect of the present disclosure includes a first storage controller and a second storage controller. The first storage controller includes a first input and output controller configured to input and output host data, and a first management controller. The second storage controller includes a second input and output controller configured to input and output host data, and a second management controller. The first management controller is configured to store a backup of software of at least one of the second storage controller or the first input and output controller. A copy of tampered software of the at least one is stored. The tampered software of the at least one is recovered by the backup.

(8) According to an aspect of the present disclosure, a storage system can be protected from tampering of software executed by the storage system.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
(1) FIG. **1** illustrates a configuration example of a storage system and related apparatus thereof according to an embodiment of the present description;
(2) FIG. **2** illustrates a configuration example of software stored by a management controller;
(3) FIG. **3** illustrates a configuration example of software stored by a disk controller;
(4) FIG. **4** illustrates a configuration example of a software information table;
(5) FIG. **5** illustrates a configuration example of a check result management table;
(6) FIG. **6** illustrates a configuration example of a heartbeat information table;
(7) FIG. **7** illustrates a configuration example of a monitoring target management table;
(8) FIG. **8** illustrates a configuration example of a log management table;
(9) FIG. **9** illustrates a configuration example of a report record management table;
(10) FIG. **10** illustrates a configuration example of a tampered firmware storage management table;
(11) FIG. **11** illustrates a flowchart of an example of a tampering inspection method in activation of a storage controller;
(12) FIG. **12** illustrates a sequence diagram of a processing example when tampering is detected in firmware of management controller;
(13) FIG. **13** illustrates an example of cooperation of two management controllers of the same storage node;
(14) FIG. **14** illustrates an example of cooperation of two management controllers of different storage nodes;
(15) FIG. **15** illustrates a flowchart of an example of the tampering inspection method in the activation of the storage controller;
(16) FIG. **16** illustrates a sequence diagram of a processing example when tampering is detected in initial phase tampering check of the disk controller; and
(17) FIG. **17** illustrates a sequence diagram of a processing example when tampering is detected in later phase tampering check of the disk controller.

DESCRIPTION OF EMBODIMENTS

(18) Hereinafter, an embodiment of the present invention will be described in detail with reference to the drawings. In the following description, the same components are denoted by the same reference numerals in principle, and repetitive description thereof is omitted. It should be noted that the embodiment described below is merely an example for implementing the present invention and does not limit the technical scope of the present invention.

(19) System Configuration

(20) Referring to FIG. **1**, a configuration example of a storage system and related apparatus thereof according to an embodiment of the present description will be described. The storage system includes one or more storage nodes **100**. FIG. **1** illustrates three storage nodes **100** as an example. One or more hosts (not illustrated) are connected to the storage nodes **100** via a network (not illustrated).

(21) In order to manage host data, each of the hosts issues various requests such as a read request or a write request (I/O request) to the storage nodes **100** via the network. The network can use a protocol such as Fibre Channel (FC) or Ethernet, for example.

(22) A management apparatus **102** is connected to each of the storage nodes **100** via a network **101**. In addition, the storage nodes **100** can communicate with each other via the network **101**.

(23) An administrator of the system manages the storage nodes **100** by operating the management apparatus **102**. The network **101** can use, for example, a local area network (LAN). As described later, when tampering of software is detected in the storage node **100**, information thereof is transmitted to the management apparatus **102**. The management apparatus **102** presents the information to the administrator in a display apparatus (not illustrated) in FIG. **1**.

(24) The management apparatus **102** may have, for example, a computer configuration. Specifically, the management apparatus **102** can include an arithmetic device, a main storage device, an auxiliary storage device, an input device, an output device, and a communication interface. The arithmetic device implements a predetermined function by executing a program stored in the main storage device. The program is, for example, loaded from the auxiliary storage device to the main storage device.

(25) In general, the main storage device includes a volatile storage medium, and the auxiliary storage device includes a non-volatile storage medium. The configuration of the management apparatus **102** is not particularly limited. The input device is, for example, a mouse, a keyboard, or a combination thereof, and the output device is, for example, a display device or a printing device.

(26) Each storage node **100** is equipped with two storage controllers (STGC) **110**A and **110**B having the same function for high system reliability. The storage node **100** may be equipped with one or more storage drives (not illustrated) as storage media for holding data from the hosts (referred to as host data). The storage drives are, for example, a hard disk drive (HDD) and a solid state drive (SSD).

(27) Hereinafter, an example of the two storage controllers **110**A and **110**B in the storage node **100** will be described. The number of storage controllers in each storage node **100** is not particularly limited. The storage node can include three or more storage controllers, and the number of equipped storage controllers may differ between the storage nodes.

(28) The storage controller **110**A includes a management controller (MGC) **120**A and a disk controller (DKC) **130**A. The above are different hardware devices. The disk controller is also referred to as an input and output controller. The storage controller **110**A further includes a management port **151**A, a host port **153**A, and an internal communication interface **155**A.

(29) The management port **151**A is an interface for the storage controller **110**A to communicate with the management apparatus **102** and a storage controller of another storage node, and the host port **153**A is an interface for the storage controller **110**A to transmit and receive the host data to and from the host.

(30) The internal communication interface **155**A is an interface for the storage controller **110**A to

communicate with the other storage controller **110**B in the same storage node. The internal communication interface **155**A stores environment information, environment setting, and the like in the system, and enables communication between the devices.

(31) The management controller **120**A manages the storage nodes **100** according to an instruction from the administrator. For example, the management controller **120**A executes setting of the storage nodes **100**, including creation and setting of a volume.

(32) The management controller **120**A includes a central processing unit (CPU) **121**A, which is an arithmetic device that executes management, a flash memory **123**A, and a plurality of SSDs **124**A and **125**A. The above are different hardware devices, and the number of each is not limited. The management controller **120**A further includes a DRAM **126**A used as a main storage device. The DRAM is a memory including a volatile storage medium.

(33) The CPU **121**A implements a predetermined management function by executing a program stored in the DRAM **126**A. The processing executed by the CPU **121**A is the processing executed by the management controller **120**A. The CPU **121**A communicates with the management apparatus **102** and other storage nodes via the management port **151**A.

(34) In an embodiment of the present description, the CPU **121**A inspects and activates software stored in and executed by itself, and inspects and activates a part of software stored in the disk controller **130**A. Inspect means to determine presence or absence of tampering of the software. For example, a known technique using an electronic signature can be used in the determination of the presence or absence of tampering of the software. Accordingly, reliability of the storage system can be improved. A subject of the inspection can be changed from the configuration in the embodiment of the present description.

(35) The flash memory **123**A and the SSDs **124**A and **125**A are storage devices with different interface protocols. For example, the flash memory **123**A can use Serial Peripheral Interface (SPI), and the SSDs **124**A and **125**A can use Non-Volatile Memory Express (NVMe). In this way, the software can be efficiently held by using different types of storage devices. In an embodiment of the present description, a capacity of each of the SSDs **124**A and **125**A is larger than a capacity of the flash memory **123**A. One type of storage device may be implemented alone.

(36) In activation of the management controller **120**A, the CPU **121**A first accesses the flash memory **123**A and activates the stored software (program). Thereafter, the CPU **121**A accesses the SSD **125**A to activate software of the SSD **125**A.

(37) The CPU **121**A inspects the software of the SSD **125**A to determine presence or absence of tampering. In addition, the CPU **121**A inspects the software of the disk controller **130**A. When tampering of the software is detected, the CPU **121**A stores the tampered software in the SSD **124**A.

(38) In this way, by storing the tampered software, detailed analysis can be performed later. In addition, as described later, the tampered software of the SSD **125**A is rolled back using a backup. Security can be improved by storing the tampered software in the SSD **124**A different from the SSD **125**A. The CPU **121**A does not need to inspect the software of the disk controller **130**A.

(39) The disk controller **130**A processes input and output of the host data. The disk controller **130**A stores the host data received from the host in the storage drive in response to the write request from the host, reads data specified from the storage drive in response to the read request from the host, and transfers the data to the host. The host data is logically stored in the volume. The volume is associated with a storage area of the storage drive.

(40) The disk controller **130**A includes a CPU **131**A, which is an arithmetic device that inputs and outputs the host data, a flash memory **133**A, and SSDs **134**A and **135**A. The above are different hardware devices. The disk controller **130**A further includes a DRAM **136**A used as a main storage device.

(41) The CPU **131**A implements a predetermined management function by executing a program stored in the main storage device. The processing executed by the CPU **131**A is the processing

executed by the disk controller **130**A. The CPU **131**A communicates with the host via the host port **153**A.

(42) The CPU **131**A inspects and activates software stored in and executed by itself. The inspection determines presence or absence of tampering of the software. Accordingly, reliability of the storage system can be improved.

(43) The flash memory **133**A and the SSDs **134**A and **135**A are storage devices with different interface protocols. For example, the flash memory **133**A can use SPI, and the SSDs **134**A and **135**A can use NVMe. In this way, the software can be efficiently held by using different types of storage devices. In one embodiment of the present description, a capacity of each of the SSDs **134**A and **135**A is larger than a capacity of the flash memory **133**A. One type of storage device may be used alone.

(44) The disk controller **130**A is activated in response to a notification from the management controller **120**A. In the embodiment of the present description, a part of the software of the disk controller **130**A is inspected by the management controller **120**A before the activation of the disk controller **130**A. When tampering is not detected, the activation of the disk controller **130**A is started.

(45) In the embodiment of the present description, the management controller **120**A inspects software stored in the flash memory **133**A. The management controller **120**A can access the flash memory **123**A not via the CPU **131**A but via the internal communication interface **155**A.

(46) After the inspection by the management controller **120**A, the disk controller **130**A accesses the flash memory **133**A to activate the inspected software, and further inspects a remaining part of the software to determine presence or absence of tampering. In the embodiment of the present description, the remaining part is software stored in the SSD **135**A. Accordingly, security reliability of the storage system can be improved.

(47) The disk controller **130**A detects tampering of the software stored in the SSD **135**A and stores the tampered software in the SSD **134**A. In this way, by storing the tampered software, detailed analysis can be performed later. The SSD **134**A stores only the tampered software without storing software for operating the storage nodes **100**. Accordingly, the security can be improved.

(48) In the embodiment of the present description, the storage controller **110**B has the same configuration as that of the storage controller **110**A, and includes components of the same type. Specifically, the storage controller **110**B includes a management controller **120**B and a disk controller **130**B. The storage controller **110**B further includes a management port **151**B, a host port **153**B, and an internal communication interface **155**B. Communication between the storage controllers **110**A and **110**B is executed via the internal communication interfaces **155**A and **155**B.

(49) Similarly to the management controller **120**A, the management controller **120**B includes a CPU **121**B, a flash memory **123**B, and SSDs **124**B and **125**B. The management controller **120**A further includes a DRAM **126**B used as a main storage device. Similarly to the disk controller **130**A, the disk controller **130**B includes a CPU **131**B, a flash memory **133**B, and SSDs **134**B and **135**B. The disk controller **130**B further includes a DRAM **136**B used as a main storage device.

(50) The CPU **121**B of the management controller **120**B can access the flash memory **133**B of the disk controller **130**B via the internal communication interface **155**B.

(51) The management controller **120**B and the disk controller **130**B respectively execute operation of the management controller **120**A and the disk controller **130**A. The management controllers **120**A and **120**B may have different configurations, and the disk controllers **130**A and **130**B may have different configurations.

(52) The storage controllers **110**A and **110**B have a redundant configuration, and when one fails, the other takes over processing to be executed by the failed one. When one of the two management controllers **120**A and **120**B fails, the other takes over the processing.

(53) As described above, the management controllers **120**A and **120**B inspect the software executed by themselves and the software of the disk controller in the same storage controller. Therefore,

when one management controller, for example, the management controller **120**B fails, the management controller **120**A of the redundant configuration executes the inspection of the software of the disk controller **130**B instead of the management controller **120**B.

(54) In the embodiment of the present description, when tampering of the software is detected in one management controller and the activation of the management controller is interrupted, normal software is transferred from the other management controller, and recovery is executed. The transfer of the normal software may be executed by the other management controller in the same node as the management controller in which tampering is detected or by another management controller in another node. Accordingly, a management controller that cannot be activated due to tampering of the software can be activated and operated.

(55) FIG. **2** illustrates a configuration example of the software (program) stored in the management controller **120**A. The management controller **120**B of the storage controller **110**B stores software similar to that of the management controller **120**A. The flash memory **123**A of the management controller **120**A stores MGC firmware **210**A. The MGC firmware **210**A includes an MGC activation program **211**A, an MGC inspection program **212**A, and an MGC monitoring program **213**A.

(56) The SSD **125**A of the management controller **120**A stores MGC firmware **220**A, an operating system (OS), and management software **230**A. The management software operates on the OS. The MGC firmware **220**A includes an MGC activation program **221**A and a file transfer program **222**A. The OS and the management software **230**A include a DKC inspection program **232**A and a DKC activation instruction program **233**A.

(57) FIG. **3** illustrates a configuration example of the software (program) stored in the disk controller **130**A. The disk controller **130**B of the storage controller **110**B stores software similar to that of the disk controller **130**A.

(58) The flash memory **133**A of the disk controller **130**A stores DKC firmware **250**A. The DKC firmware **250**A includes a DKC initial phase activation program **251**A and a DKC inspection program **252**A. The SSD **135**A of the disk controller **130**A stores the DKC firmware **250**A. DKC firmware **260**A includes a DKC later phase activation program **261**A. The DKC firmware **260**A includes, for example, software that inputs and outputs the host data.

(59) Management Information

(60) FIGS. **4** to **10** illustrate tables included in management information held by the management controllers **120**A and **120**B. In these tables, management controllers and disk controllers having a redundant configuration are held in the system as common information. The management information is stored in, for example, SSDs of the management controllers and the disk controllers.

(61) FIG. **4** illustrates a configuration example of a software information table **310**. The software information table **310** indicates version information of software to be executed by the management controllers and the disk controllers in the system. Device column **311** indicates a type of the controller, that is, the management controller or the disk controller. Location ID column **312** identifies a storage controller in which the management controller or the disk controller is present. SW ver. column **313** indicates a version of the software executed by the management controller or the disk controller.

(62) FIG. **5** illustrates a configuration example of a check result management table **320**. The check result management table **320** stores an inspection result of the software of the storage controller. Date column **321** indicates a date and time of the inspection, and Device column **322** indicates a type of the controller that is an inspection target, that is, the management controller or the disk controller. Location ID column **323** identifies a storage controller in which the management controller or the disk controller is present. Result column **324** indicates the inspection result.

(63) FIG. **6** illustrates a configuration example of a heartbeat information table **330**. A monitoring result between the management controllers is stored. Date column **331** indicates a date and time at which a heartbeat signal is received, and MGC Location ID column **332** indicates a transmission

source of the heartbeat signal.

(64) FIG. **7** illustrates a configuration example of a monitoring target management table **340**. The monitoring target management table **340** indicates a disk controller that is a management target and the other management controller in cooperation. MGC ID column **341** indicates an ID of the management controller. DKC ID column **342** indicates a disk controller managed by the management controller indicated by MGC ID column **341**. MGC ID column **343** indicates an ID of a management controller that is a monitoring target, which cooperates with the management controller indicated by MGC ID column **341**. In the example, the IDs of the disk controller and the management controller coincide with their Location IDs.

(65) FIG. **8** illustrates a configuration example of a log management table **350**. The log management table **350** manages event logs generated in the system. The log management table **350** includes log information when tampering occurred. Date column **351** indicates an event occurrence date and time. Event ID column **352** indicates an ID of an occurred event. Event Name column **353** indicates a name of the occurred event. Severity column **354** indicates importance of the event. Device column **355** indicates a device in which the event occurred. Location ID column **356** indicates an ID of a storage controller in which the event occurred.

(66) FIG. **9** illustrates a configuration example of a report record management table **360**. The report record management table **360** stores a report record when tampering occurred. A report is transmitted to, for example, the management apparatus **102**.

(67) Date column **361** indicates an occurrence date and time of an event that is a reporting target. Event ID column **362** indicates an ID of the target event. Event Name column **363** indicates a name of the target event. Severity column **364** indicates importance of the target event. Device column **365** indicates a device in which tampering is detected. Location ID column **366** indicates an ID of a storage controller in which tampering is detected.

(68) FIG. **10** illustrates a configuration example of a tampered firmware storage management table **370**. The tampered firmware storage management table **370** manages tampered firmware stored in a dedicated SSD.

(69) Date column **371** indicates a date and time when the tampered firmware is stored. Device column **372** indicates a device in which tampering is detected. Location ID column **373** indicates an ID of a storage controller in which tampering is detected. Backup Device column **374** indicates a device in which the tampered firmware is stored. Here, a controller storage that are stored are illustrated. Backup Location ID column **375** indicates an ID of a storage controller in which the tampered firmware is stored. In the example, each management controller and each disk controller include a storage device that stores only one tampered firmware.

(70) Tampering Detection of Software of Management Controller

(71) FIG. **11** illustrates a flowchart of an example of a tampering inspection method in the activation of the storage controller **110**A. The storage controller **110**B also executes the same processing. In the processing example described with reference to FIG. **11**, when tampering is detected in firmware of a management controller, recovery thereof is executed. In addition, the tampered firmware is stored in a dedicated SSD of the management controller.

(72) First, the activation of the management controller **120**A is started (S**11**). The CPU **121**A of the management controller **120**A checks tampering of the software stored in the management controller **120**A (S**12**). Specifically, the CPU **121**A activates the MGC activation program **221**A of the MGC firmware **220**A stored in the SSD **125**A starting from execution of the MGC activation program **211**A stored in the flash memory **123**A.

(73) Further, the CPU **121**A executes the MGC inspection program **212**A in the MGC firmware **210**A stored in the flash memory **123**A. The CPU **121**A executes inspection of the OS and the management software **230**A stored in the SSD **125**A according to the MGC inspection program **212**A. An inspection result is recorded in the check result management table **320**.

(74) When tampering is detected in the OS and the management software **230**A stored in the

management controller **120**A (S**12**: NG), since reliability of the management controller **120**A is reduced, the activation of the storage controller **110**A is stopped. Accordingly, higher reliability can be obtained.

(75) When tampering is not detected (S**12**: OK), the CPU **121**A activates the OS and the management software **230**A according to the MGC activation program **221**A (S**13**). The management software is activated after the OS.

(76) Next, the CPU **121**A inspects an unexecuted program of the MGC firmware **220**A according to the MGC inspection program **212**A. An inspection result is recorded in the check result management table **320**. When tampering is not detected (S**14**: OK), the CPU **121**A executes the inspected program of the MGC firmware **220**A (S**15**).

(77) When tampering is detected in the MGC firmware **220**A (S**14**: NG), the CPU **121**A stores the tampered firmware in the SSD **124**A of the management controller **120**A according to the MGC inspection program **212**A, and registers information thereof in the tampered firmware storage management table **370** (S**16**).

(78) Next, the CPU **121**A executes recovery of the tampered firmware according to the MGC inspection program **212**A (S**17**). Specifically, the CPU **121**A refers to the software information table **310** to identify a software version of the management controller **120**A, and updates the MGC firmware **220**A in which tampering is detected by a backup of MGC firmware having the version. The backup is stored in, for example, the SSD **125**A or the flash memory **123**A of the management controller **120**A.

(79) The CPU **121**A further records a tampering log indicating the execution of the recovery in the log management table **350** according to the MGC inspection program **212**A (S**18**). Subsequently, the CPU **121**A reports the management apparatus **102** in which tampering is detected and the recovery is executed according to the MGC inspection program **212**A (S**19**). Report information is recorded in the report record management table **360**. Thereafter, the flow returns to step S**14**.

(80) FIG. **12** illustrates a sequence diagram of a processing example when tampering is detected in the MGC firmware (FW) **220**A. In the following description of each sequence diagram, reception and execution of a data access instruction to the flash memory or the SSD and response of an execution result are actually processed by the firmware or the software stored in the flash memory or the SSD, but the description thereof is omitted as processing by the flash memory or the SSD.

(81) The MGC activation program (PG) **211**A of the management controller **120**A activates another program of the MGC firmware **210**A in the flash memory **123**A by secure boot (S**31**). The MGC activation program **211**A further activates the MGC activation program **221**A of the MGC firmware **220**A in the SSD **125**A by the secure boot (S**32**).

(82) When initial activation of the firmware **210**A and **220**A of the management controller **120**A is completed (S**33**), the MGC inspection program **212**A inspects the OS and the management software **230**A in the SSD **125**A (S**34**). When an inspection result indicates that there is no tampering (S**35**), the MGC activation program **221**A activates the OS in the OS and the management software **230**A (S**36**), and then activates the management software (S**37**).

(83) Next, the MGC inspection program **212** inspects the unexecuted software in the MGC firmware **220**A in the SSD **125**A (S**38**). When tampering is detected (S**39**), the MGC inspection program **212**A copies tampered firmware **240** in the SSD **125**A from the SSD **125**A to the SSD **124**A (S**41**, S**42**, and S**43**). The MGC inspection program **212**A instructs the SSD **125**A to copy tampered firmware **240**A to the SSD **124**A. The SSD **125**A receives a completion response from the SSD **124**A, and returns a copy success response to the MGC inspection program **212**A.

(84) Next, the MGC inspection program **212**A recovers the tampered firmware (S**44**, S**45**, and S**46**). For example, the MGC inspection program **212** instructs the SSD **125**A to update the tampered firmware by a backup stored in the SSD **125**A. The MGC inspection program **212**A instructs the SSD **125** to update, and the SSD **125** returns a success response after the update.

(85) Next, an example of cooperation of two management controllers will be described. When one

management controller is normally activated and then the other management controller cannot be normally activated, in the embodiment of the present description, normal firmware is copied from the normally activated management controller, and firmware of the management controller where the activation is failed is rolled back. Safety can be improved by recovery by a backup from the other management controller.

(86) Management controllers monitored by each management controller are registered in the monitoring target management table **340**. In addition, each controller can monitor the other management controller by using the heartbeat signal. A monitoring result obtained by the heartbeat signal is registered in the heartbeat information table **330** as described above.

(87) FIG. **13** illustrates an example of cooperation of two management controllers storage node. A management controller activated first confirms activation of the other management controller. When the activation of the firmware of the other management controller cannot be detected for a predetermined time, the normally activated management controller copies normal firmware to the other management controller, and rolls back the firmware of the management controller where the activation is failed. The management controller can execute processing described below in addition to or instead of the processing described with reference to FIG. **12**.

(88) In an example described below, the management controller **120**B fails to activate, and the normal MGC firmware is copied from the normally activated management controller **120**A to the management controller **120**B. As described with reference to FIG. **12**, it is assumed that the management controller **120**B checks tampering of the OS and the management software and then detects tampering in the inspection of the MGC firmware **220**. The management controller **120**B stores the tampered firmware in the SSD **124**B. The backup is not stored in the management controller **120**B, or the rollback cannot be performed by the backup and the activation is interrupted.

(89) After the management controller **120** is normally activated (S**36**, S**37**), the MGC monitoring program **213**A of the management controller **120**A repeats activation confirmation of the management controller **120**B for a predetermined period of time (S**51**). When there is no normal activation response from an MGC monitoring program **213**B of the management controller **120**B (S**52**), the MGC monitoring program **213**A instructs the file transfer program **222**A of the management controller **120**A to transfer the normal MGC firmware (S**53**).

(90) The file transfer program **222**A reads the normal MGC firmware stored in the SSD **125**A of the management controller **120**A (S**54**), and transfers the normal MGC firmware to the management controller **120**B (S**55**).

(91) A file transfer program **222**B of the management controller **120**B receives the normal MGC firmware from the management controller **120**A and stores the normal MGC firmware in the SSD **125**B of the management controller **120**B (S**56**). MGC firmware **220**B in the SSD **125**B may be rewritten with the transferred MGC firmware.

(92) Upon receiving a completion response from the SSD **125**B (S**57**), the file transfer program **222**B returns a transfer completion notification to the file transfer program **222**A (S**58**). The file transfer program **222**A notifies the MGC monitoring program **213**A of success of the normal MGC firmware (S**59**).

(93) The MGC monitoring program **213**A instructs an MGC inspection program **212**B of the management controller **120**B to re-inspect the MGC firmware **220**B (S**60**). The MGC inspection program **212**B re-inspects the MGC firmware **220**B (S**61**), and obtains a normal inspection result (S**62**). After the reinspection, the management controller **120**B executes subsequent activation. The security is improved by the reinspection.

(94) The cooperation described with reference to FIG. **13** is executed between the management controllers of the same storage node. The same processing can be executed between management controllers of different storage nodes as illustrated in FIG. **14**. After activation of one storage node is completed, a management controller of the storage node whose activation is completed confirms

activation of a management controller of another storage node.

(95) In a configuration example in FIG. **14**, the management controller **120**A of a storage node **100**A confirms activation of a management controller **120**C of the other storage node **100**B. The management controller **120**C includes a CPU **121**C and an SSD **125**C, which correspond to the CPU **121**B and the SSD **125**B of the management controller **120**B in FIG. **13**. In addition, the CPU **121**C executes an MGC management program **213**C, an MGC inspection program **212**C, and a file transfer program **222**C. The above correspond to the MGC monitoring program **213**B, the MGC inspection program **212**B, and the file transfer program **222**B of the management controller **120**B in FIG. **13**. MGC firmware **220**C stored in the SSD **125**C corresponds to the MGC firmware **220**B stored in the SSD **125**B. When the activation of the MGC firmware by the management controller **120**C of the storage node **100**B cannot be detected for a predetermined time, the normally activated management controller **120**A copies the normal MGC firmware to the management controller **120**C where the activation is failed. Therefore, the management controller **120**C that received the normal MGC firmware rolls back the MGC firmware.

(96) After completion of the rollback of the MGC firmware, the management controller **120**C re-inspects the MGC firmware. The storage node **100**B that copied and recovered the MGC firmware from the other storage node **100**A executes cooperation between management controllers in the same storage node **100**B as described with reference to FIG. **13**.

(97) Tampering Detection of Software of Disk Controller

(98) Next, tampering detection of software of a disk controller will be described. FIG. **15** illustrates a flowchart of an example of a tampering inspection method in the activation of the storage controller **110**A. In FIG. **15**, steps S**71**, S**72**, and S**73** are the same as S**11**, S**12**, and S**13** of the flowchart illustrated in FIG. **11**. In addition, in FIG. **15**, steps S**14** to S**19** illustrated in FIG. **11** are omitted, and the management controller **120**A may execute steps S**14** to S**19** illustrated in FIG. **11** also in the present processing.

(99) In step S**74**, the CPU **121**A of the management controller **120**A executes DKC initial phase tampering check according to the DKC inspection program **232**A. Specifically, the CPU **121**A executes the DKC inspection program **232**A, accesses the flash memory **133**A of the disk controller **130**A via the internal communication interfaces **155**A and **155**B, and inspects the DKC firmware **250**A. An inspection result is recorded in the check result management table **320**.

(100) When tampering of the DKC firmware **250**A is not detected (S**74**: OK), DKC initial phase activation is started (S**75**). Specifically, the CPU **121**A instructs the CPU **131**A of the disk controller **130**A to activate according to the DKC activation instruction program **233**A. The CPU **131**A activates the DKC initial phase activation program **251**A of the inspected DKC firmware **250**A. The CPU **131**A activates a program including the DKC inspection program **252**A according to the DKC initial phase activation program **251**A.

(101) The CPU **131**A executes DKC later phase tampering check (S**76**). Specifically, the CPU **131**A inspects tampering of the DKC firmware **260**A stored in the SSD **135**A according to the DKC inspection program **252**A. An inspection result is recorded in the check result management table **320**.

(102) When tampering of DKC firmware **260**A is not detected (S**76**: OK), the CPU **131**A sequentially activates the programs of the DKC firmware **260**A according to the DKC later phase activation program **261**A (DKC later phase activation). Accordingly, the activation of the storage controller **110**A is completed.

(103) Referring to step S**74**, when tampering is detected in the DKC initial phase tampering check (S**74**: NG), the CPU **121**A of the management controller **120**A stores the tampered DKC firmware (S**77**). Specifically, the CPU **121**A stores the tampered DKC firmware **250**A in the SSD **124**A of the management controller **120**A according to the DKC inspection program **232**A. Accordingly, detailed analysis of the tampered firmware can be performed later.

(104) Further, the CPU **121**A executes recovery of the disk controller **130**A according to the DKC

inspection program **232**A (S**78**). Specifically, the CPU **121**A refers to the software information table **310** to identify a software version of the disk controller **130**A, and updates the DKC firmware **250**A in which tampering is detected by a backup of DKC firmware having the version. The backup is stored, for example, in the SSD **135**A or the flash memory **133**A of the disk controller **130**A.

(105) The CPU **121**A further records a tampering log indicating the execution of the recovery in the log management table **350** according to the DKC inspection program **232**A (S**79**). Subsequently, the CPU **121**A reports the management apparatus **102** in which tampering is detected and the recovery is executed according to the DKC inspection program **232**A (S**80**). Report information is recorded in the report record management table **360**. Thereafter, the flow returns to step S**74**. When tampering is detected by the recovered software, the recovery is executed by another backup software or the activation is stopped.

(106) Referring to step S**76**, when tampering is detected in the DKC later phase tampering check (S**76**: NG), the CPU **131**A of the disk controller **130**A stores the tampered DKC firmware (S**77**). Specifically, the CPU **121**A stores the tampered DKC firmware **260**A in the SSD **124**A of the management controller **120**A according to the DKC inspection program **232**A. Accordingly, detailed analysis of the tampered firmware can be performed later.

(107) Further, the CPU **131**A executes the recovery of the disk controller **130**A according to the DKC inspection program **252**A (S**82**). The CPU **131**A refers to the software information table **310** to identify the software version of the disk controller **130**A, and updates the tampered DKC firmware **260**A by the backup of the DKC firmware having the version. The backup is stored, for example, in the SSD **125**A of the management controller **120**A or the SSD **135**A of the disk controller **130**A.

(108) The CPU **131**A further records a tampering log indicating the execution of the recovery in the log management table **350** according to the DKC inspection program **252**A (S**83**). Subsequently, the CPU **131**A reports the management apparatus **102** in which tampering is detected and the recovery is executed according to the DKC inspection program **252**A (S**84**). Report information is recorded in the report record management table **360**. Thereafter, the flow returns to step S**76**.

(109) FIG. **16** is a sequence diagram of a processing example when tampering is detected in the initial phase tampering check S**74**. The DKC inspection program **232**A of the management controller **120**A inspects the DKC firmware **250**A stored in the flash memory **133**A of the disk controller **130**A (S**101**). When an inspection result indicates that there is a tampering (S**102**), the DKC inspection program **232**A copies the tampered DKC firmware from the flash memory **133**A to the SSD **124**A of the management controller **120**A (S**103**). Data on the management controller **120**A at the time of subsequent analysis is easily collected. The tampered firmware may be stored in the SSD **134**A of the disk controller **130**A.

(110) When the copy is completed (S**104**), the DKC inspection program **232**A rolls back the DKC firmware **250**A of the flash memory **133**A according to backup firmware stored in the SSD **135**A of the disk controller **130**A (S**105** and S**106**). The DKC inspection program **232**A may instruct the flash memory **133**A to roll back by a backup of the SSD **135**A, or the CPU **131**A of the disk controller **130**A that received a request from the DKC inspection program **232**A may update the DKC firmware **250**A of the flash memory **133**A.

(111) The flash memory **133**A returns the completion of the rollback of the DKC firmware **250**A to the DKC inspection program **232**A (S**108**).

(112) In the above example, a backup of the DKC firmware **250**A is held in the SSD **135**A of the disk controller **130**A. In another example, the backup may be stored in the SSD or the flash memory of the management controller **120**A.

(113) FIG. **17** is a sequence diagram of a processing example when tampering is detected in the later phase tampering check S**76**. The DKC initial phase activation program **251**A turns on power of the disk controller **130** (S**121**). In addition, the DKC initial phase activation program **251**A

sequentially activates other programs of the DKC firmware **250**A in the flash memory **133**A (S**122**).

(114) The activated DKC inspection program **252**A inspects the DKC firmware **260**A in the SSD **135**A in response to an inspection instruction (S**123**) from the DKC initial phase activation program **251**A (S**124**). When an inspection result indicates that there is a tampering (S**125**), the DKC inspection program **252**A instructs the SSD **135** of the disk controller **130**A to copy the tampered firmware to the SSD **134** (S**126**). The SSD **135**A copies the tampered firmware to the SSD **134**A (S**127**), and receives a completion response (S**12**).

(115) When the DKC inspection program **252**A receives, from the SSD **135**A, the copy completion response of the tampered firmware (S**129**), the DKC inspection program **252**A rolls back the DKC firmware **260**A of the SSD **135** of the disk controller by a backup of the SSD **125** of the management controller **120**A.

(116) Specifically, the DKC inspection program **252**A instructs the SSD **125**A to copy the backup of the DKC firmware **260**A (S**130**). The SSD **125**A copies the backup of the DKC firmware **260**A to the SSD **135**A (S**131**). Upon receiving the copy completion response from the SSD **135**A (S**132**), the SSD **125**A returns a rollback completion response of the DKC firmware **260**A to the DKC inspection program **252**A (S**133**). The safety can be improved by using a backup of another controller.

(117) The storage controller may execute periodic tampering check (inspection). Reliability after the activation of the storage controller can be maintained by the periodic tampering check. For example, the periodic tampering check is executed every time a predetermined period of time elapses after the activation of the storage controller.

(118) A method of the tampering check is basically the same as the tampering check at the time of the activation described above. In the periodic check, the management controller executes processing independently of the input and output of the host data by the disk controller while the disk controller is operating. The input and output of the host data by the disk controller can be executed in parallel with the tampering check and the recovery by the management controller.

(119) The management controller may take over processing of another management controller in which a fault occurred. For example, upon detecting a fault of the management controller **120**B, the management controller **120**A takes over the processing of the management controller **120**B. A plurality of management controllers that can take over processing of one management controller may exist, and one management controller may take over processing of a plurality of fault management controllers.

(120) The management controller **120**A monitors a heartbeat signal from a management controller that is a monitoring target. Monitoring targets are illustrated, for example, in the monitoring target management table **340**. The heartbeat signal can always monitor other management controllers.

(121) The present invention is not limited to the above embodiment and includes various modifications. For example, the above embodiment has been described in detail for easy understanding of the present invention, and the present invention is not necessarily limited to those including all the configurations described above. A part of a configuration of one embodiment can be replaced with a configuration of another embodiment, and a configuration of one embodiment can be added to a configuration of another embodiment. In addition, a part of a configuration of each embodiment can be added, deleted, or replaced with another configuration.

(122) Each of the above configurations, functions, processing units, or the like may be partially or entirely implemented by hardware such as design using an integrated circuit. In addition, each of the above configurations, functions, and the like may be implemented by software through means of a processor interpreting and executing a program for implementing respective functions. Information of programs, tables, files, or the like for implementing each function can be placed in a recording apparatus such as a memory, a hard disk, and a solid state drive (SSD), or a recording medium such as an IC card, an SD card, or the like.

(123) Control lines and information lines show those considered to be necessary for the description, and not all the control lines and the information lines are necessarily shown on the product. Actually, it may be considered that almost all the configurations are connected to each other.

Claims

1. A storage system comprising: a first storage controller; and a second storage controller, wherein the first storage controller includes a first input and output controller configured to input and output host data, and a first management controller, the second storage controller includes a second input and output controller configured to input and output host data, and a second management controller, the first management controller is configured to store a backup of software of at least one of the second storage controller or the first input and output controller, a copy of tampered software of the at least one is stored, and the tampered software of the at least one is recovered by the backup.

2. The storage system according to claim 1, wherein the tampered software is stored in a storage drive different from a storage drive storing the software before the tampering.

3. The storage system according to claim 1, wherein the backup is a backup of the software of the second storage controller.

4. The storage system according to claim 3, wherein the second storage controller is included in a storage node different from the first storage controller.

5. The storage system according to claim 3, wherein the backup is a backup of the software of the second management controller, and the first management controller is configured to recover the tampered software by the backup when normal activation of the second management controller cannot be confirmed.

6. The storage system according to claim 5, wherein the second management controller is configured to inspect the recovered software.

7. The storage system according to claim 1, wherein the backup is a backup of the software of the first input and output controller, and the first input and output controller is configured to, when tampering of the software is detected, store the tampered software in a storage drive of the first input and output controller, and recover the tampered software by the backup.

8. A method for inspecting tampering of software in a storage system, wherein the storage system includes a first storage controller, and a second storage controller, the first storage controller includes a first input and output controller configured to input and output host data, and a first management controller, the second storage controller includes a second input and output controller configured to input and output host data, and a second management controller, the first management controller is configured to store a backup of software of at least one of the second storage controller or the first input and output controller, and the method comprises: by the storage system, inspecting the software of the at least one; storing a copy of the software of the at least one in which tampering is detected by the inspection; and recovering the tampered software of the at least one by the backup.