ALTERNATE PARTS SIGNATURE LIST FILE

BACKGROUND

The present disclosure relates generally to aircraft and in a particular to a method and apparatus for managing software on an aircraft. Still more particularly, the present disclosure relates to a computer implemented method, apparatus, and computer usable program code for managing software or aircraft parts on an aircraft.

Modern aircraft are extremely complex. For example, an aircraft may have many types of electronic systems on board. An electronic system on an aircraft may also be referred to as a line replaceable unit (LRU). Each line replaceable unit may further take on various forms. A line replaceable unit may be, for example, without limitation, a flight management system, an autopilot, an in flight entertainment system, a communications system, a navigation system, a flight controller, a flight recorder, and a collision avoidance system.

Line replaceable units such as these may use software or programming to provide the logic or control for various operations and functions. The software used in these line replaceable units is also commonly treated as parts in the airline industry. In particular, a software application for use in a line replaceable unit on an aircraft may also be tracked separately and referred to as a loadable software aircraft part (LSAP) or software aircraft parts.

When an entity, such as an airline, receives an aircraft, software aircraft parts are typically already installed on the line replaceable units in the aircraft. In some cases, the software aircraft parts may need to be reinstalled or reloaded onto the line replaceable units in the aircraft. The installation or reloading of software may be time consuming and tedious.

Aircraft software parts are beginning to be distributed electronically to reduce the time and effort required for distributing aircraft software parts. Digital signatures are used to secure the aircraft software parts. Since aircraft signatures and their corresponding certificates expire, there is a need to update the signature validity onboard the aircraft from time to time.

Thus, it would be advantageous to have an improved method and apparatus for distributing software aircraft parts and updating their digital signatures that addresses the above described problems and others.

SUMMARY

Advantageous embodiments provide a computer implemented method, apparatus, and computer program product for managing software aircraft parts. In one embodiment, a computer implemented method creates an alternate parts signature list file having a set of signatures for a set of software aircraft parts. The alternate parts signature list file is signed, and the alternate parts signature list file is placed in a crate to form a crated alternate parts signature list file. The crated alternate parts signature list file signed to form a signed crate. The signed crate is sent to an aircraft data processing system. Each signature for a software aircraft part on the aircraft data processing system is replaced with a corresponding signature in the signed crate to form a set of current signatures on the aircraft data processing system.

In another advantageous embodiment, a set of software aircraft parts is identified. A set of signatures associated with the set of software aircraft parts is placed in a file, and the file is sent to an aircraft data processing system as a software aircraft part.

In a further advantageous embodiment, a software aircraft part containing a set signatures associated with a set of software aircraft parts is received on an aircraft. Each old signature associated with a set of software aircraft parts is replaced with a software aircraft part in the set of software aircraft parts on the aircraft with the corresponding new signature in the alternate parts signature list.

In another advantageous embodiment, an apparatus comprises an aircraft data processing system in an aircraft and a software application located on the aircraft data processing system. The software application is capable of receiving a software aircraft part having a set of signatures associated with a set of software aircraft parts in the aircraft data processing system. The software application is also capable of using a selected signature in the set of signatures in a file to authenticate a selected software aircraft part in the set of software aircraft parts stored on the aircraft data processing system to form an authenticated software aircraft part, and loading the authenticated software part onto a line replaceable unit in the aircraft data processing system.

In yet another advantageous embodiment, a computer program product comprises a computer recordable media and a program code stored on the computer recordable media. The program code includes code for identifying a set of software

aircraft parts and for placing a set of signatures associated with the set of software aircraft parts in a file. The program code also includes code for sending the file an aircraft data processing system as a software aircraft part.

The features, functions, and advantages can be achieved independently in various embodiments of the present disclosure or may be combined in yet other embodiments in which further details can be seen with reference to the following description and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the advantageous embodiments are set forth in the appended claims. The advantageous embodiments, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an advantageous embodiment of the present disclosure when read in conjunction with the accompanying drawings, wherein:

Figure 1 is a pictorial representation of a network of data processing systems in which the advantageous embodiments may be implemented;

Figure 2 is a diagram of a data processing system in accordance with an illustrative embodiment;

Figure 3 is a diagram illustrating a mechanism for re-signing aircraft parts in accordance with an advantageous embodiment;

Figure 4 is a diagram of a software aircraft parts management system in accordance with an advantageous embodiment;

Figure 5 is a diagram of an aircraft data processing system and line replaceable units in accordance with an advantageous embodiment;

Figure 6 is a diagram of a valid part list in accordance with an advantageous embodiment;

Figure 7 is a diagram illustrating an alternate parts signature list file in accordance with an advantageous embodiment;

Figure 8 is a diagram illustrating an alternate parts signature list file in the form of a software aircraft part in accordance with an advantageous embodiment;

Figure 9 is a diagram illustrating a structure of an alternate parts signature list file in accordance with an advantageous embodiment;

Figure 10 is a flowchart of a process for managing software aircraft parts in accordance with an advantageous embodiment;

Figure 11 is a flowchart for process for managing software aircraft parts on an aircraft in accordance with an advantageous embodiment;

Figure 12 is a flowchart of a process for receiving an alternate parts signature list file in accordance with an advantageous embodiment;

Figure 13 is a flowchart of a process for loading a software aircraft part in accordance with an advantageous embodiment; and

Figure 14 is a flowchart of a process for processing an alternate parts signature list in accordance with an advantageous embodiment.

DETAILED DESCRIPTION

With reference now to the figures and in particular with reference to **Figures 1-2**, exemplary diagrams of data processing environments are provided in which advantageous embodiments may be implemented. It should be appreciated that **Figures 1-2** are only exemplary and are not intended to assert or imply any limitation with regard to the environments in which different embodiments may be implemented. Many modifications to the depicted environments may be made.

Figure 1 depicts a pictorial representation of a network of data processing systems in which the advantageous embodiments of the present invention may be implemented. Network data processing system 100 is a network of computers in which embodiments may be implemented. Network data processing system 100 contains network 102, which is the medium used to provide communications links between various devices and computers connected together within network data processing system 100. Network 102 may include connections, such as wire, wireless communication links, or fiber optic cables.

In the depicted example, server **104** and server **106** connect to network **102** along with storage unit **108**. In addition, clients **110**, **112**, and **114** connect to network **102**. These clients **110**, **112**, and **114** may be, for example, personal computers or network computers. In the depicted example, server **104** provides data, such as boot files, operating system images, and applications to clients **110**, **112**, and **114**.

Clients **110**, **112**, and **114** are clients to server **104** in this example. Aircraft **116** also is a client that may exchange information with clients **110**, **112**, and **114**. Aircraft

116 also may exchange information with servers **104** and **106**. Aircraft **116** may exchange data with different computers through a wireless communications link while in-flight or any other type of communications link while on the ground. In these examples, advantageous embodiments may be implemented in network data processing system **100** to manage software aircraft parts on an aircraft, such as aircraft **116**. In particular, the different advantageous embodiments provide a computer implemented method, apparatus, and computer program product for managing signatures used to authenticate software aircraft parts on aircraft **116**. Network data processing system **100** may include additional servers, clients, and other devices not shown.

In the depicted example, network data processing system **100** is the Internet with network **102** representing a worldwide collection of networks and gateways that use the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols to communicate with one another. Of course, network data processing system **100** also may be implemented as a number of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). **Figure 1** is intended as an example, and not as an architectural limitation for different embodiments.

Turning now to **Figure 2**, a diagram of a data processing system is depicted in accordance with an illustrative embodiment. Data processing system **200** is an example of a data processing system that may be used to implement servers and clients, such as server **104** and client **110.** Further, data processing system **200** is an example of a data processing system that may be found in aircraft **116** in **Figure 1**.

In this illustrative example, data processing system **200** includes communications fabric **202**, which provides communications between processor unit **204**, memory **206**, persistent storage **208**, communications unit **210**, input/output (I/O) unit **212**, and display **214**.

Processor unit **204** serves to execute instructions for software that may be loaded into memory **206**. Processor unit **204** may be a set of one or more processors or may be a multi-processor core, depending on the particular implementation. Further, processor unit **204** may be implemented using one or more heterogeneous processor systems in which a main processor is present with secondary processors on a single chip.

As another illustrative example, processor unit **204** may be a symmetric multi-processor system containing multiple processors of the same type.

Memory **206**, in these examples, may be, for example, a random access memory or any other suitable volatile or non-volatile storage device. Persistent storage **208** may take various forms depending on the particular implementation. For example, persistent storage **208** may contain one or more components or devices. For example, persistent storage **208** may be a hard drive, a flash memory, a rewritable optical disk, a rewritable magnetic tape, or some combination of the above. The media used by persistent storage **208** also may be removable. For example, a removable hard drive may be used for persistent storage **208**.

Communications unit **210,** in these examples, provides for communications with other data processing systems or devices. In these examples, communications unit **210** is a network interface card. Communications unit **210** may provide communications through the use of either or both physical and wireless communications links.

Input/output unit **212** allows for input and output of data with other devices that may be connected to data processing system **200**. For example, input/output unit **212** may provide a connection for user input through a keyboard and mouse. Further, input/output unit **212** may send output to a printer. Display **214** provides a mechanism to display information to a user.

Instructions for the operating system and applications or programs are located on persistent storage **208**. These instructions may be loaded into memory **206** for execution by processor unit **204**. The processes of the different embodiments may be performed by processor unit **204** using computer implemented instructions, which may be located in a memory, such as memory **206**. These instructions are referred to as, program code, computer usable program code, or computer readable program code that may be read and executed by a processor in processor unit **204**. The program code in the different embodiments may be embodied on different physical or tangible computer readable media, such as memory **206** or persistent storage **208**.

Program code **216** is located in a functional form on a selectively removable computer readable media **218** and may be loaded onto or transferred to data processing system **200** for execution by processor unit **204**. Program code **216** and computer readable media **218** form computer program product **220** in these examples. In one example, computer readable media **218** may be in a tangible form, such as, for

example, an optical or magnetic disc that is inserted or placed into a drive or other device that is part of persistent storage **208** for transfer onto a storage device, such as a hard drive that is part of persistent storage **208**. In a tangible form, computer readable media **218** also may take the form of a persistent storage, such as a hard drive, a thumb drive, or a flash memory that is connected to data processing system **200**. The tangible form of computer readable media **218** is also referred to as computer recordable storage media. In some instances, the media 218 may not be removable.

Alternatively, program code **216** may be transferred to data processing system **200** from computer readable media **218** through a communications link to communications unit **210** and/or through a connection to input/output unit **212**. The communications link and/or the connection may be physical or wireless in the illustrative examples. The computer readable media also may take the form of non-tangible media, such as communications links or wireless transmissions containing the program code.

The different components illustrated for data processing system **200** are not meant to provide architectural limitations to the manner in which different embodiments may be implemented. The different illustrative embodiments may be implemented in a data processing system including components in addition to or in place of those illustrated for data processing system **200**. Other components shown in **Figure 2** can be varied from the illustrative examples shown.

For example, a bus system may be used to implement communications fabric **202** and may be comprised of one or more buses, such as a system bus or an input/output bus. Of course, the bus system may be implemented using any suitable type of architecture that provides for a transfer of data between different components or devices attached to the bus system. Additionally, a communications unit may include one or more devices used to transmit and receive data, such as a modem or a network adapter. Further, a memory may be, for example, memory **206** or a cache such as found in an interface and memory controller hub that may be present in communications fabric **202**.

The different advantageous embodiments may be used with software aircraft parts that are signed with digital signatures. A software aircraft part may be signed with a private key. A public key for the software aircraft part is stored in the aircraft. The private and public key combinations are used to determine whether the software aircraft

part is from the expected or authorized source. The public key is used to verify that the sender signed the software aircraft part and that the software aircraft part has not been modified. A modification may be due to the corruption of the software aircraft part or through tampering.

In the different advantageous embodiments, the public key is stored on the aircraft data processing system to authenticate software aircraft parts before they are loaded onto line replaceable units within the aircraft. This process is referred to as an authentication process. In other words, this private key/public key pair may be used to establish that the software aircraft part is authentic.

In the advantageous embodiments, when an aircraft is received by an airline, the public key changes from a manufacturer public key to an airline pubic key. As a result, the software aircraft parts currently on the aircraft can no longer be authenticated. In these situations, an airline may reload software aircraft parts with the software aircraft parts that have been signed with the new private key. The software aircraft parts are reloaded onto a storage device on the aircraft in case they are needed at a later point in time.

Software aircraft parts may sometimes be present on an aircraft for ten or more years before being changed. Keys are typically only valid for around two years before they expire. One manner in which the software aircraft parts may be reinstalled when signatures expire is to sign each software aircraft part with the new private key and transfer all of the software aircraft parts onto the aircraft data processing system for storage. The public key corresponding to the new private key is sent to the aircraft data processing system. This type of process is expensive and time consuming and may cause delays in the maintenance of an aircraft.

The different advantageous embodiments recognize that this type of software aircraft part management on aircraft requires an improved method and apparatus for managing signatures to reduce the amount of time and effort needed to reload the software aircraft parts. In other words, the different advantageous embodiments recognize that it would be useful to avoid redistributing all software aircraft parts onto an aircraft data processing system if signatures for these parts expire.

If the signatures for the software aircraft parts have expired or are no long valid for some other reason, the software aircraft parts have to be reloaded onto the line replaceable units in the aircraft. With this situation, the software aircraft parts have to

be re-signed, or signed again, with a new valid private key. As used herein, the terms resign, resigned, or resigning are meant to convey an act of providing a new signature. The software aircraft parts with the new signatures are then reloaded into storage on the aircraft. A new public key corresponding to the private key is loaded onto the aircraft data processing system. The software aircraft parts can then reinstalled onto the line replaceable units in the aircraft if needed.

Thus, it would be advantageous to have an improved method and apparatus for distributing software aircraft parts that solves the above described problems.

Turning now the **Figure 3**, a diagram illustrating a mechanism for resigning aircraft parts is depicted in accordance with an advantageous embodiment. In this example, software aircraft parts management system **300** is a software management system for an entity, such as an airline, a maintenance repair and overhaul provider (MRO), or a military squadron. Software aircraft parts management system **300** may be located on a ground data processing system, such as sever **104** in **Figure 1,** on which software aircraft parts may be managed and stored.

Aircraft data processing system **302**, in these examples, is a set of computers on an aircraft, such as aircraft **116** in **Figure 1**. As used in the different illustrative examples, "a set" refers to one or more items. For example, a set of computers is one or more computers, and a set of signatures is one or more signatures. Aircraft data processing system **302** contains software aircraft parts with signatures.

Instead of re-distributing each software aircraft part with a new signature, the different advantageous embodiments send the new signatures without the software aircraft parts. The signatures are generated from the software aircraft parts stored on the ground, and the signatures are used on the aircraft data processing system to authenticate the software aircraft parts on the aircraft.

In these examples, the software aircraft part is alternate parts signature list file **304**. This file contains the signatures for software aircraft parts on aircraft data processing system **302**. By placing a list of the new signatures for each of the aircraft parts on aircraft data processing system **302** into alternate parts signature list file **304**, only this file is loaded onto aircraft data processing system **302**, instead of all of the software aircraft parts. For example, an aircraft fleet may have over 500 software aircraft parts, and an aircraft may have 200 or more software aircraft parts that require signatures.

Alternate parts signature list file **304** contains the new signatures for each of the software aircraft parts. These new signatures may replace the current signatures stored on aircraft data processing system **302** for each of the software aircraft parts. As a result, when a software aircraft part is to be loaded onto a line replaceable unit on an aircraft, a valid signature is present allowing the software aircraft part to be authenticated before loading onto a line replaceable unit. A signature may expire if the signature is too old or past an expiration date. This situation results in an invalid signature. An invalid signature may also be present if, for example, without limitation, ownership of the aircraft changes. Signatures for the previous owner become invalid even though the signature may not have expired.

Turning now to **Figure 4**, a diagram of a software aircraft parts management system is depicted in accordance with an advantageous embodiment. In this example, software aircraft parts management system **400** is an example of one implementation of software aircraft parts management system **300** in **Figure 3**.

In this example, software aircraft parts management system **400** includes software aircraft parts library **402,** management process **404**, and user interface **406**. These software aircraft parts may be managed through management process **404**. A user may operate management process **404** through user interface **406**, in these examples. Management process **404** may be used to distribute software aircraft parts to different aircraft.

Management process **404** may include processes to receive a software aircraft part for inclusion within software aircraft parts library **402**. Further, management process **404** includes processes or code used to authenticate software aircraft parts within software aircraft parts library **402**. Further, management process **404** also may sign the different software aircraft parts within software aircraft parts library **402** prior to the distribution of these software aircraft parts to various aircraft managed by software aircraft parts management system **400**.

In these examples, the signing process includes inputting the software aircraft part and a private key into a signing algorithm that generates the signature for the software aircraft part as a result of the inputs. The different advantageous embodiments may use any signing algorithm or process that is available. The particular algorithm or process used depends on the particular implementation.

Additionally, in the advantageous embodiments, management process **404** may be used to generate alternate parts signature list file **408.** Alternate parts signature list file **408** may contain signatures for one more software aircraft parts within software aircraft parts library **402**. In the depicted embodiments, these signatures are "detached signatures" because they are not integrated as part of the software aircraft parts. This file may take the form of a software aircraft part for loading onto an aircraft data processing system, in these examples.

Alternate parts signature list file **408** also may have a digital signature to indicate the authenticity of the part that contains this file. In these examples, alternate parts signature list file **408** may be an extensible markup language (XML) data structure containing the signature. An extensive markup language signature, in these examples, may be used to sign data resources, such as software aircraft parts library **402**.

When signatures expire or are no longer valid for some other reason, an operator may select a set of software aircraft parts from software aircraft parts library **402** for signature generation in the alternate parts signature list file **408**. In other embodiments, all of the software aircraft parts within software aircraft parts library **402** may have signatures generated and placed into alternate parts signature list file **408**. Valid software aircraft parts are parts with valid signatures. Alternate parts signature list file **408** may then be sent to an aircraft for use in replacing expired signatures for software aircraft parts on the aircraft. In the depicted examples, alternate parts signature list file **408** is sent in the form of a software aircraft part to the aircraft data processing system. This form is used to have the aircraft data processing system receive and store the file, containing these signatures, as a software aircraft part. This part also is signed, like other software aircraft parts.

Turning now to **Figure 5**, a diagram of an aircraft data processing system and line replaceable units is depicted in accordance with an advantageous embodiment. In this example, aircraft data processing system **500** and line replaceable units **502** are located on an aircraft. line replaceable units **502** are different data processing systems on the aircraft that perform various functions. For example, line replaceable units **502** may include, for example, an in-flight entertainment system, an auto-pilot, a flight management system, a hydraulics system, or an air conditioning unit.

In this example, aircraft data processing system **500** includes **504**, storage device **506**, and data load function **510**. Onboard electronic distribution system **504**

may store public keys **508**. Storage device **506** stores software aircraft parts **516** and signatures **518**. In these examples, signatures **518** are separate data structures from software aircraft parts **516**. In these examples, signatures **518** take the form of extensible mark up language (XML) signatures.

In these examples, onboard electronic distribution system **504** performs authentication of the software aircraft parts when the parts are received and just before the parts are loaded on to line replaceable units **502**. Onboard electronic distribution system **504** is used to authenticate a software aircraft part in software aircraft parts **516** before a data load function loads the software aircraft part onto line replaceable units **502**.

Onboard electronic distribution system **504** performs authentication processes on a software part in software aircraft parts **516** to determine whether that software can be loaded onto a line replaceable unit in line replaceable units **502** by data load function **510**. If a signature corresponding to the aircraft part to be loaded onto line replaceable units **502** has expired or is invalid for some other reason, data load function **510** will not load that software aircraft part onto line replaceable units **502**.

In the different advantageous embodiments, alternate parts signature list (APSL) file **520** may be loaded as a software aircraft part, such as software aircraft part **522**, by onboard electronic distribution system **504** and stored in storage device **506**. Signatures within alternate parts signature list file **520** may be used to replace invalid signatures within signatures **518**. As a result, software aircraft parts with expired signatures do not have to be re-distributed to aircraft data processing system **500**.

This type of feature is especially advantageous when signatures expire for multiple software aircraft parts within software aircraft parts **516**. Further, when a new aircraft is delivered, software aircraft parts **516** do not have to be re-distributed in their entirety with the use of alternate parts signature list file **520** to replace an invalid signature in signatures **518**.

In other advantageous embodiments, instead of replacing signatures **518**, onboard electronic distribution system **504** may check a current signature within signatures **518**. If the current signature expired, then onboard electronic distribution system **504** may check alternate parts signature list file **520** to see whether the corresponding new signature is present in this software aircraft part. If the corresponding new signature is present in alternate parts signature list file **520**, onboard

electronic distribution system **504** may perform the authentication with that new signature.

Turning now to **Figure 6**, a diagram of a valid part list file is depicted in accordance with an advantageous embodiment. Valid part number list file **600** contains a list of parts that are to be included in an alternate parts signature list file. In this example, parts **602**, **604**, **606**, and **608** are present in valid part number list file **600**. This file may be created through a user selecting parts located in a software aircraft parts library. Alternatively, valid part number list file **600** may include all of the part numbers present in the library. In yet other embodiments, this file may contain all part numbers within a particular type of aircraft or class of aircraft.

With reference now to **Figure 7**, a diagram illustrating an alternate parts signature list file is depicted in accordance with an advantageous embodiment. In this example, alternate parts signature list file **700** includes an identification of parts **602**, **604**, **606**, and **608** from valid part number list file **600** in **Figure 6**. This file is an example of alternate parts signature list file **304** is **Figure 3**, alternate parts signature list file **408** in **Figure 4**, and alternate parts signature list file **520** in **Figure 5**. In addition, signatures **702**, **704**, **706**, and **708** are new valid signatures generated for these different parts.

With reference now to **Figure 8**, a diagram illustrating an alternate parts signature list file in the form of a software aircraft part is depicted in accordance with an advantageous embodiment. In this example, software aircraft part **800** includes alternate parts signature list file **802** and header file **804**. Alternate parts signature list file **802** is a file, such as alternate parts signature list file **700** in **Figure 7**.

Header file **804**, in these examples, is a header file that identifies alternate parts signatures list file **802** as a loadable software aircraft part. In these examples, the header is defined through a standard promulgated by Aeronautical Radio, Incorporated (ARINC). The standard used is an ARINC 665, in these examples. In particular, the software aircraft part is an airline modifiable information (AMI) software aircraft part. This part also includes information to identify software aircraft part file **800** as being a part that can be loaded onto the aircraft data processing system.

Turning now to **Figure 9**, a diagram illustrating a structure of an alternate parts signature list file is depicted in accordance with an advantageous embodiment. In this example, alternate parts signature list file **900** includes unique identifier **902**, date/time

904, source **906**, alternate parts signature list file part number **908**, and signature **910**. signature **910** includes part number **912** and signature body **914**.

Unique identifier **902** is a unique identifier that identifies alternate parts signature list file **900** from other lists or software aircraft parts, in these examples. Date/time **904** is the date and time when alternate parts signature list file **900** was created. Source **906** identifies the source that created alternate parts signature list file **900**. In these examples, the source may be, for example, a particular user or system.

Alternate parts signature list file part number **908** identifies the alternate parts signature list file as a particular aircraft part. Part number **912** contains the part number of the software aircraft part. Signature body **914** is an encoded signature extensive markup language file, in these examples. Of course, signature body **914** may use other types of signatures other than extensive markup language signatures.

Turning next to **Figure 10**, a flowchart of a process for managing software aircraft parts is depicted in accordance with an advantageous embodiment. The process illustrated in **Figure 10** may be implemented in a software aircraft part management system, such as software aircraft parts management system **400** in **Figure 4** in the illustrative example. Of course, these steps may be implemented in other data processing systems or computer devices, depending on the particular implementation.

The process begins by identifying a set of software aircraft parts (operation **1000**). This set of software aircraft parts may be identified in a number of different ways. For example, a user may select the set of software aircraft parts from a software aircraft parts library. Alternatively, the set of software aircraft parts may be identified by a computer implemented process. For example, management process **404** in **Figure 4** may identify some or all software aircraft parts within software aircraft parts library **402** in **Figure 4** without using input to form a set of software aircraft parts. The identification may be made based on a set of rules. For example, a rule may specify selecting the software aircraft parts as any software aircraft part that is for a particular model or manufacturer of aircraft.

Thereafter, a set of signatures associated with the set of software aircraft parts are placed into a file (operation **1002**). In these examples, each signature in the file associated with a particular software aircraft part is a signature generated using the software aircraft part. In these examples, the digital signature is created using a private

key. A public key located on the aircraft is used to authenticate the signature. In other words, the public key may be used to determine whether the software aircraft part associated with the signature is actually from the recorded source and/or unmodified.

The process then creates a software aircraft part with the file containing the signatures (operation **1004**). In these examples, the software aircraft part may be created from the file containing the signatures for the software aircraft parts by adding or associating a header or header file, such as header file **804** in **Figure 8**. By creating a software aircraft part from the file containing the digital signatures, the digital signatures may be transmitted to an aircraft data processing system as part of a normal process of transferring software aircraft parts. As a result, special or different procedures do not have to be created to introduce the signatures onto the aircraft data processing system. The aircraft data processing system recognizes these digital signatures as being a software aircraft part for purposes of receiving and storing the digital signatures.

The process signs the part (operation **1006**). In these examples, the signing is performed using a private key. A public key may later be used in an authentication to prove that the part is authentic. Thereafter, the signed part is placed into a crate (operation **1008**). In these examples, a crate is a file using a file format that may store a set of files, such as software aircraft parts. The set of files in the crate may be compressed to reduce their size. In other words, a crate acts as a wrapper for one or more items. An example of a crate system is a ZIP file format. Another example is a Java® archive file format. These types of files also may be used as crates in a different advantageous embodiment.

Thereafter, the process signs the crate (operation **1010**). Operations **1004**-**1010** are optional steps that may be used in some, but not all of the advantageous embodiments. At this point, a signed crate is ready for transmission to the aircraft data processing system. The process sends the part to an aircraft data processing system on an aircraft (operation **1012**).

In operation **1012**, the part may be sent to the aircraft data processing system in a number of different ways. For example, a wireless communications link may be used to send the part to the aircraft data processing system. With this type of communications link, the aircraft may be located in various locations, such as, for example, a runway, a boarding terminal, a maintenance facility, or the air.

Alternatively, the file may be sent to the aircraft data processing system through a physical means. For example, an operator may carry a storage device containing the file and connect the storage device to the aircraft data processing system to transfer the file to the aircraft data processing system. This storage device may be, for example, a compact disc, a flash memory, or a hard drive on a laptop computer.

Turning now to **Figure 11**, a flowchart of a process for managing software aircraft parts on an aircraft is depicted in accordance with an advantageous embodiment. The process illustrated in **Figure 11** may be implemented in an aircraft data processing system, such as aircraft data processing system **500** in **Figure 5**. In particular, the different illustrative processes may be implemented in a software component, such as onboard electronic distribution system **504** and data load function **510** in **Figure 5**.

The process begins by receiving the file containing the set of signatures (operation **1100**). In these examples, the file is received in the form of a software aircraft part. Of course, in other embodiments, the file may be received in a format other than as a software aircraft part, depending on the particular implementation. As an example, the file may be received as an extensible mark up language file in other embodiments.

The file is stored (operation **1102**). The process then loads a software aircraft part on the aircraft using a signature in the file corresponding to the software aircraft part (operation **1104**). The process then terminates. This operation occurs when a software aircraft part stored on the aircraft data processing system is to be loaded onto a line replaceable unit within the aircraft. The digital signature corresponding to the software aircraft part is used in place of the old digital signature, which may have expired or become invalid for other reasons.

With reference now to **Figure 12**, a flowchart of a process for receiving an alternate parts signature list file is depicted in accordance with an advantageous embodiment. The process illustrated in **Figure 12** may be implemented in an aircraft data processing system, such as aircraft data processing system **500** in **Figure 5**. In particular, the different processes may be implemented using onboard electronic distribution system **504** in **Figure 5**.

The process begins by receiving a crate containing the alternate parts signature list file as a software aircraft part (operation **1200**). The process authenticates the signatures on the crate and the software aircraft part (operation **1202**). This

authentication may be performed, in these examples, by sending a request to controller server module **508** in **Figure 5**. In response to this request, controller server module **508** in **Figure 5** uses onboard authentication service **511** in **Figure 5** and the appropriate public key in public keys **512** in **Figure 5** to determine whether the crate and the software aircraft part in the crate are authentic.

A signature may be authenticated using a signature verifying algorithm. This type of algorithm uses the signature or the signed file and a public key corresponding to the private key used to generate the signature. The algorithm used will depend on the particular implementation.

A determination is made as to whether the authentication is successful (operation **1204**). If the authentication is successful, the process stores the alternate parts signature list file in storage on the aircraft (operation **1204**). In this example, the storage may be storage device **506** in **Figure 5**.

The process then processes the alternate parts signature list file for use (operation **1206**), with the process terminating thereafter. In operation **1206**, the signatures within the alternate parts signature list file may be used to replace old signatures corresponding to the software aircraft parts.

In other embodiments, the alternate parts signature list file may be stored and maintained without replacing signatures. In this type of embodiment, if an authentication process fails with an existing signature, the alternate parts signature list file is consulted to determine whether a signature is present for the software aircraft part that failed authentication. If the signature is present, then that signature is used to perform the authentication. With reference again to operation **1204**, if the authentication is not successful, the process terminates.

With reference now to **Figure 13**, a flowchart of a process for loading a software aircraft part is depicted in accordance with an advantageous embodiment. The process illustrated in **Figure 13** may be implemented in an aircraft data processing system, such as aircraft data processing system **500** in **Figure 5**. In particular, the process may be implemented using data load function **510** in **Figure 5**.

The process begins by receiving a request to load a software aircraft part onto a line replaceable unit (operation **1300**). In these examples, a request is initiated by a human operator. Of course, in other embodiments, the request may be initiated through

other mechanisms. For example, an event, such as a message sent from an airline server to the aircraft data processing system may be used to initiate this process.

The digital signature for a software aircraft part is located (operation **1302**). Authentication is performed on the software aircraft part using the located digital signature (operation **1304**). A determination is made as to whether the authentication is successful (operation **1306**). If the digital signature is no longer valid, the authentication is not successful.

In this case, the alternate parts signature list file is located (operation **1308**). A determination is made as to whether a signature for the software aircraft part is present in the alternate parts signature list file (operation **1310**). If the signature is present, the process returns to operation **1304** to perform authentication using this located signature. The authentication may also fail for other reasons. For example, if the digital signatures and/or software aircraft part becomes corrupted, the authentication will fail, in these examples. Otherwise, an error is generated (operation **1312**), with the process terminating thereafter.

With reference again to operation **1306**, if the authentication of the software aircraft part is successful, the software aircraft part is loaded from storage onto a line replaceable unit (operation **1314**), with the process terminating thereafter.

Turning now to **Figure 14**, a flowchart of a process for processing an alternate parts signature list is depicted in accordance with an advantageous embodiment. In this example, the process illustrated in **Figure 14** may be implemented in an aircraft data processing system, such as aircraft data processing system **500** in **Figure 5**. In particular, the process may be implemented using onboard electronic distribution system **504** in **Figure 5**.

The process begins by receiving a request to process an alternate parts signature list (operation **1400**). A determination is made as to whether an alternate parts signature list is available for processing (operation **1402**). If an alternate parts signature list is available, a determination is made as to whether a universal identifier (UID) file is available (operation **1404**).

If a universal identifier file is available, a determination is made as to whether the universal identifier value in the universal identifier file matches the universal identifier value in the alternate parts signature list (operation **1406**). If the universal identifier value in the file does not match the universal identifier value in the alternate parts

signature list, the process reads the part number in the alternate parts signature list file (operation **1408**).

A determination is made as to whether the part exists in the onboard storage in the aircraft data processing system (operation **1410**). The determination in this example is made using the part number in the alternate parts signature list file and comparing that number with additional parts on the aircraft data processing system. If the part exists on the aircraft data processing system, the process replaces the part signature with the signature from the alternate parts signature list corresponding to the part number (operation **1412**).

Thereafter, the process determines whether the part number in the alternate parts signature list file is the last part number in the file (operation **1414**). If the part number is the last part number, the process writes the universal identifier value from the alternate parts signature list to the universal identifier file (operation **1416**), with the process terminating thereafter.

With reference again to operation **1414**, if the part number is not the last part number in the alternate parts signature file list, the process returns to operation **1408** to read another part number from the alternate parts signature list file. With reference back to operation **1410**, if the part does not exist in the aircraft data processing system, the process continues to operation **1414** as described above.

Turning back to operation **1406**, if the universal identifier value in the file matches the universal identifier value in the alternate parts signature list, the process terminates. With reference again to operation **1404**, if a universal identifier file is unavailable, the process proceeds to operation **1408** as described above. With reference back to operation **1402**, if an alternate parts signature list is unavailable, the process terminates.

The flowcharts and block diagrams in the different depicted embodiments illustrate the architecture, functionality, and operation of some possible implementations of apparatus, methods and computer program products. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of computer usable or readable program code, which comprises one or more executable instructions for implementing the specified function or functions. In some alternative implementations, the function or functions noted in the block may occur out of the order noted in the figures. For example, in some cases, two blocks shown in succession may

be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved.

Thus, the different advantageous embodiments provide a computer implemented method, apparatus, and computer useable code for managing software aircraft parts. In the different advantageous embodiments, a set of software aircraft parts are identified, a set of signatures associated where these set of software aircraft parts are placed in the file. The file is then sent to the aircraft data processing system. The signatures in the file may be used to authenticate the set of software aircraft parts that are located on the aircraft. In this manner, reinstalling each software aircraft part for which a digital signature has expired is avoided.

Instead, with some or all of the different advantageous embodiments, only the digital signatures need to be sent to the aircraft. This reduction in the amount of information sent to the aircraft saves both bandwidths in a network as well as time and expense that are currently needed to reinstall each software aircraft part with the new digital signature.

The different advantageous embodiments can take the form of an entirely hardware embodiment, an entirely software embodiment, or an embodiment containing both hardware and software elements. Some embodiments are implemented in software, which includes but is not limited to forms, such as, for example, firmware, resident software, and microcode.

Furthermore, the different embodiments can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any device or system that executes instructions. For the purposes of this disclosure, a computer-usable or computer readable medium can generally be any tangible apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

The computer usable or computer readable medium can be, for example, without limitation an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, or a propagation medium. Non limiting examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a

rigid magnetic disk, and an optical disk. Optical disks may include compact disk – read only memory (CD-ROM), compact disk – read/write (CD-R/W) and DVD.

Further, a computer-usable or computer-readable medium may contain or store a computer readable or usable program code such that when the computer readable or usable program code is executed on a computer, the execution of this computer readable or usable program code causes the computer to transmit another computer readable or usable program code over a communications link. This communications link may use a medium that is, for example without limitation, physical or wireless.

A data processing system suitable for storing and/or executing computer readable or computer usable program code will include one or more processors coupled directly or indirectly to memory elements through a communications fabric, such as a system bus. The memory elements may include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some computer readable or computer usable program code to reduce the number of times code may be retrieved from bulk storage during execution of the code.

Input/output or I/O devices can be coupled to the system either directly or through intervening I/O controllers. These devices may include, for example, without limitation to keyboards, touch screen displays, and pointing devices. Different communications adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Non-limiting examples are modems and network adapters are just a few of the currently available types of communications adapters.

The description of the different advantageous embodiments has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the embodiments in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. For example, although the new signatures in the alternate parts signature list file files in these examples are turned into or created in the form of an aircraft part for distribution onto an aircraft, other formats may be used. For example, the signatures may be sent as part of a data file rather then a software aircraft part depending on the particular implementation. Further, these processes illustrated in the advantageous embodiments may be applied to other types

of vehicles or devices. For example, the different advantageous embodiments may be applied to vehicles, such as, for example, without limitation, submarines, surface ships, cars, a spacecraft in which software parts may be reloaded onto different data processing systems from storage located on the vehicles. Further, the different advantageous embodiments may be applied to any devices that store software parts that may be reloaded onto different systems in which authentication is required for reloading.

Further, different advantageous embodiments may provide different advantages as compared to other advantageous embodiments. The embodiment or embodiments selected are chosen and described in order to best explain the principles of the embodiments, the practical application, and to enable others of ordinary skill in the art to understand the disclosure for various embodiments with various modifications as are suited to the particular use contemplated.

CLAIMS:

What is claimed is:

1. A computer implemented method for managing software aircraft parts, the computer implemented method comprising:

creating an alternate parts signature list file having a set of signatures for a set of software aircraft parts;

signing the alternate parts signature list file;

placing the alternate parts signature list file in a crate to form a crated alternate parts signature list file;

signing the crated alternate parts signature list file to form a signed crate;

sending the signed crate to an aircraft data processing system as a software aircraft part; and

replacing each signature for a software aircraft part on the aircraft data processing system with a corresponding signature in the signed crate to form a set of current signatures on the aircraft data processing system.

2. The computer implemented method of claim 1, wherein the sending step comprises:

sending the signed crate to an onboard electronic distribution system on the aircraft data processing system.

3. The computer implemented method of claim 1 further comprising:

validating a particular software aircraft part prior to loading onto a line replaceable unit using a corresponding current signature for the set of current signatures.

4. The computer implemented method of claim 3, wherein the loading step is initiated by a human operator.

5. A computer implemented method for managing software aircraft parts, the computer implemented method comprising:

identifying a set of software aircraft parts;

placing a set of signatures associated with the set of software aircraft parts in a file; and

sending the file as a software aircraft part to an aircraft data processing system, in which the set of signatures correspond to existing software aircraft parts.

6. The computer implemented method of claim 5 further comprising:

storing the software aircraft part in the aircraft data processing system;

using a selected signature in the set of signatures in the a software aircraft part file to authenticate a selected software aircraft part in the set of software aircraft parts stored on the aircraft data processing system to form an authenticated software aircraft part; and

loading the authenticated software part onto a line replaceable unit in the aircraft.

7. The computer implemented method of claim 5, wherein the loading step is initiated by a human operator.

8. The computer implemented method of claim 5 further comprising:

replacing each old signature for a set of software aircraft parts on the aircraft data processing system with a corresponding new signature located in the software aircraft part.

9. The computer implemented method of claim 5, wherein the each old signature is an expired signature.

10. The computer implemented method of claim 5, wherein the placing step comprises:

placing the set of signatures associated with the set of software aircraft parts in the file;

creating the software aircraft part from the file;

signing the software aircraft part to form a signed software aircraft part;

placing the signed software aircraft part in a crate to form a crated software aircraft part;

11. The computer implemented method of claim 5, wherein the software aircraft part is stored in a crate.

12. The computer implemented method of claim 5, wherein the set of signatures is a set of extensible markup language signatures.

13. A computer implemented method for managing software aircraft parts, the computer implemented method comprising:

receiving a software aircraft part containing a set of signatures associated with a set of software aircraft parts on an aircraft; and

replacing each old signature associated with each software aircraft part in the set of software aircraft parts on the aircraft with a corresponding new signature in the software aircraft part .

14. The computer implemented method of claim 13 further comprising:

placing the set of signatures associated with the set of software aircraft parts in a file;

forming the software aircraft part with the file; and

sending the software aircraft part to an aircraft data processing system.

15. The computer implemented method of claim 13, wherein the replacing step comprises:

responsive to a request to load a selected software aircraft part in the set of software aircraft parts on the aircraft, authenticating the selected software aircraft part using an associated new signature from the software aircraft part in place of an old signature associated with the selected software aircraft part;

responsive to the selected software aircraft part being successfully authenticated, loading the selected software aircraft part onto a line replaceable unit on the aircraft.

16. The computer implemented method of claim 15, wherein the request is sent by an aircraft mechanic.

17. An apparatus comprising:

an aircraft data processing system in an aircraft; and

a software application located on the aircraft data processing system, wherein the software application is capable of receiving a software aircraft part having a set of signatures associated with a set of software aircraft parts in the aircraft data processing system and using a selected signature in the set of signatures in a file to authenticate a selected software aircraft part in the set of software aircraft parts stored on the aircraft data processing system to form an authenticated software aircraft part, and loading the authenticated software part onto a line replaceable unit in the aircraft.

18. The apparatus of claim 17 further comprising:

a second data processing system; and

a software aircraft part management application located on the second data processing system, wherein the software aircraft part management application is capable of identifying the set of software aircraft parts; placing the set of signatures associated with the set of software aircraft parts in the file; and sending the file to the aircraft data processing system as a software aircraft part.

19. A computer program product comprising:

a computer recordable media;

program code, stored on the computer recordable media, for identifying a set of software aircraft parts;

program code, stored on the computer recordable media, for placing a set of signatures associated with the set of software aircraft parts in a file to form a software aircraft part; and

program code, stored on the computer recordable media, for sending the software aircraft part to an aircraft data processing system.

20. The computer program product of claim 19 further comprising:

program code, stored on the computer recordable media, for storing the software aircraft part in the aircraft data processing system;

program code, stored on the computer recordable media, for using a selected signature in the set of signatures in the software aircraft part to authenticate a selected software aircraft part in the set of software aircraft parts stored on the aircraft data processing system to form an authenticated software aircraft part; and

program code, stored on the computer recordable media, for loading the authenticated software part onto a line replaceable unit in the aircraft data processing system.