(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 March 2013 (14.03.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/036733 A1

(51) International Patent Classification:
G06F 15/16 (2006.01)

(21) International Application Number:
PCT/US2012/054108

(22) International Filing Date:
7 September 2012 (07.09.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/532,527 8 September 2011 (08.09.2011) US
13/308,363 30 November 2011 (30.11.2011) US

(71) Applicant *(for all designated States except US)*: **LEXMARK INTERNATIONAL, INC** [US/US]; IP Law Department/ Bldg. 082-1, 740 West New Circle Road, Lexington, KY 40550 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ADKINS, Christopher, Alan** [US/US]; 4904 Waynes Boulevard, Lexington, KY 40513 (US). **RADEMACHER, Timothy, John** [US/US]; 1713 Shenandoah Drive, Lexington, KY 40504 (US).

(74) Agent: **ESSER, William, F**; Lexmark International, Inc., IP Law Department/ Bldg. 082-1, 740 West New Circle Road, Lexington, KY 40550 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: SYSTEM AND METHOD FOR SECURED HOST-SLAVE COMMUNICATION

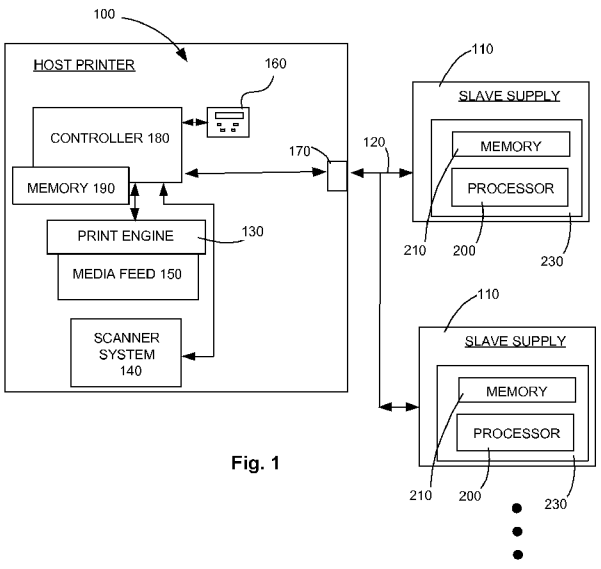

Fig. 1

(57) Abstract: A new system and method for communicating between a host device and one or more slave devices are presented. The system provides data error checking and correcting, data encryption, and robust slave address sequencing using a portion of a session key. The data encryption uses a second portion of the session key, which changes for each power cycle.

WO 2013/036733 A1

SYSTEM AND METHOD FOR SECURED HOST-SLAVE COMMUNICATION

CROSS REFERENCES TO RELATED APPLICATIONS

[0001] The present application is related to and claims priority from U.S. provisional patent application 61/532,527, filed September 8, 2011, entitled, "SYSTEM AND METHOD FOR SECURED MASTER-SLAVE COMMUNICATION," the content of which is incorporated by reference herein its entirety.

BACKGROUND

[0002] 1. Field of Disclosure

[0003] Example embodiments of the present disclosure relate generally to secure master-slave communication, and more particularly to a communication system and method in which a session key is generated by both the master and slave devices for use in both encryption/decryption and slave address generation.

[0004] 2. Description of the Related Art

[0005] Printing devices are known to use electronic authentication schemes associated with their consumable supply items. Typically, the replaceable supply item contains an integrated circuit chip that communicates with the controller located in the printer. In such an arrangement, the printer is configured as the host device and each supply item as a slave device. The controller in the host checks the authenticity of each slave device by sending a challenge thereto. The authenticity is verified by the host receiving from the slave device the correct response to the challenge.

[0006] In some existing consumable authentication schemes, the host and slave devices communicate over the I^2C bus. The host sends commands to the slave using the slave address assigned thereto, the slave executes the commands and sends responses, as appropriate, back to the host. The commands and data are sent with no data checking.

[0007] While the communications between hosts and slaves are not encrypted, such a system utilizes a unique slave address change feature in order to make duplicating the function of the slave device more difficult. The slave address is changed on a regular basis to slave address values determined by an algorithm that is known to both the host

and slave. After receiving an address change command from the host, the slave will not respond to address polls from the host until after a certain command is received on the new address. The current address is stored in non-volatile memory of both the host and slave so the current address, along with the position in the address sequence, is maintained over power cycles.

[0008] The address change feature makes cloning the integrated circuit chip of the slave device more difficult because the algorithm for computing the next slave address value utilizes the current value thereof. The problem with this feature is the host and slave can become unsynchronized in the address sequence. For example, this will happen when moving a slave supply item from one host printer to another because the second printer will not know where the slave device is in the address sequence. To overcome this, a means for resetting the sequence is provided, which substantially weakens the security of the system.

[0009] In particular, the existing system suffers from 1) a lack of data checking and correcting; 2) unencrypted communication; and 3) resettable slave address sequences.

[0010] Operation in noisy environments may cause data corruption on the bus, but the existing system does not have means for detecting or correcting these noise induced errors. This is of some importance because the supply items (slave devices) are often located within the host printer a relatively long distance from the host controller and the communications bus wires may be routed near aggressive noise sources, such as motors. Sending the commands in unencrypted form allows an attacker to learn the system's commands and data by capturing traffic between the printer controller and the supply item.

[0011] Based upon the foregoing, a need exists for an improved host-slave communication system.

SUMMARY

[0012] Example embodiments overcome shortcomings with existing communication schemes and thereby satisfy a significant need for a slave device for

securely communicating with a host over a bus. The slave device may include a processor and memory coupled thereto having stored therein program code instructions. The stored program code instructions, when executed by the processor, cause the processor to: following the slave device being reset, determine a seed value based upon a seed value of the slave device prior to the slave device being reset; receive a host number from a host that is substantially random; determine a session key based upon the determined seed value and the host number, the session key being substantially random; and use the session key to perform encryption and decryption operations on data to be transmitted and data received by the slave device, respectively, and to determine an address value for the slave device for communicating with the host. By creating a session key that is not communicated with the host and which is used in encryption/decryption as well as slave address generation, the slave device cooperates with the host for securely communicating therewith.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The above-mentioned and other features and advantages of the various embodiments, and the manner of attaining them, will become more apparent and will be better understood by reference to the accompanying drawings.

[0014] Figure 1 is a block diagram of a communication system including a host device and at least one slave device; and

[0015] Figure 2 is a flowchart illustrating an operation of the slave device of Figure 1 according to an example embodiment.

DETAILED DESCRIPTION

[0016] It is to be understood that the invention is not limited in its application to the details of construction and the arrangement of components set forth in the following description or illustrated in the drawings. The invention is capable of other embodiments and of being practiced or of being carried out in various ways. Also, it is to be understood that the phraseology and terminology used herein is for the purpose of description and should not be regarded as limiting. The use of "including," "comprising," or "having" and variations thereof is meant herein to encompass the items

listed thereafter and equivalents thereof as well as additional items. Unless otherwise limited, the terms "connected," "coupled," and variations thereof herein are used broadly and encompass direct and indirect connections and couplings. In addition, the terms "connected" and "coupled" and variations thereof are not restricted to physical or mechanical connections or couplings. Furthermore, and as described in subsequent paragraphs, the specific mechanical configurations illustrated in the drawings are intended to exemplify embodiments of the invention and that other alternative mechanical configurations are possible.

[0017] Example embodiments of the present disclosure are directed to communication between a host device 100 and one or more slave devices 110, as shown in Fig. 1. Host device 100 and slave device 110 communicate over a bus 120. In an example embodiment, host device 100 is a printing device and slave device 110 is a replaceable supply item. In particular, host device 100 may include components and modules typically utilized in printers, including a print engine 130 for imparting an image onto a sheet of media. For example, print engine 130 may be a print engine for a laser printer or for an inkjet printer. It is understood that print engine 130 may be any engine used in creating an image onto a sheet of media. Host device 100 may further include a scanner system 140 for capturing an image appearing on a media sheet for subsequent use in a printing operation, email communication or the like. A media feed system 150 may be included in host device 100 to successively move sheets of media from an input stack (not shown) to print engine 130 for performing a printing operation after which the printed sheet may be moved to an output area of host device 100 (not shown). The details of print engine 130, scanner system 140 and media feed system 150 are well known and will not be described herein for reasons of simplicity.

[0018] Host device 100 may further include a user interface 160 which allows for communication between host device 100 and a user thereof. User interface 160 may be any interface for facilitating communication between host device 100 and the user, such as, for example, a touch screen.

[0019] Host device 100 may further include an interface port 170 for communicating with one or more slave devices 110 over bus 120. Host device 100 may

further include a controller 180 for controlling the different components of host device 100. In the context in which host device 100 is a printing device, controller 180 may control the operation of print engine 130, scanner system 140, media feed system 150, user interface 160 and interface 170. Controller 180 may execute instructions stored in memory 190 in order to control the various components of host device 100.

[0020] In an embodiment in which host device 100 is a printing device, slave device 110 may be an ink or toner cartridge or bottle, for example. In addition or in the alternative, slave device 110 may be another replaceable component of a host laser printer, such as a developer unit of print engine 130 or a fuser unit.

[0021] Slave device may include a processor 200 for, among other things, cooperating with host device 100 in performing slave authentication so as to only allow authorized slave devices to communicate with host device 100 and thereby prevent attacks on or damage to host device 100. Processor 200 is coupled to memory 210 having instructions stored therein for execution by processor 200. Processor 200 and memory 210 may be formed in an integrated circuit chip 230. In an alternative embodiment, processor 200 and memory 210 reside in separate integrated circuit chips. In still another alternative embodiment, slave device 110 may include circuitry, such as state machine based circuitry, for cooperating with host device 100 in performing slave authentication.

[0022] It is understood that host device 100 is not limited to a printing device and may be virtually any electronics device to which a removable and/or replaceable item may communicate over bus 120. It is similarly understood that slave device 110 may be virtually any replaceable item which communicates with host device 100, including slave devices which are communicatively coupled thereto on a temporary basis.

[0023] Bus 120 may be any bus which supports a bus protocol in which a host 100 and one or more slave devices 110 communicate with each other. According to an example embodiment, bus 120 may be an Inter-Integrated Circuit (I^2C) bus. In an I^2C bus, one wire of the shared bus 120 carries data in a bidirectional manner, and another wire carries clock signals from the host device 100 to the slave devices 110. Also, while

the shared bus 120 is illustrated as a two-wire serial bus, shared parallel bus structures can be utilized.

[0024] According to at least some embodiments, including embodiments in which bus 120 is an I^2C bus, bus 120 is a master-slave bus, with host device 100 serving as the bus master and slave devices 110 as the bus slaves. When using the I^2C protocol, the host device 100 initiates all communications with the respective slave devices 110. The slave devices 110 only respond to the requests of the host device 100. In the event that an imposter is connected to the shared bus 120 and employs a valid slave address, then the imposter device can receive a communication directed to it from the host device 100. When sensitive information is passed on the bus 120 to the slave devices 110, the imposter device can receive the same in an unauthorized manner, unknown to the host device 100. This can occur if an authorized slave device 110 were to be unplugged from the shared bus 120 and the imposter device plugged therein and programmed or wired to assume the address of the slave device 110 that was unplugged. If the slave devices 110 were all equipped with fixed addresses, which has been the established practice, then it is not overly complicated to couple an imposter device to the shared bus 120 and receive sensitive communications in an unauthorized manner unknown to the host device 100. As a result, slave devices 110 occasionally change their slave addresses in response to a request by host device 100.

[0025] In an example embodiment, the host 100 and slave 110 communicate using commands and data encrypted using a stream cipher or other encryption scheme. Establishing an encryption session is done by exchanging values between the host 100 and slave 110. Then the host 100 and slave 110 each independently calculates a session key from exchanged values and a secret that is known to both. The session key is then used to initialize the cipher (or other encryption scheme) and the slave address function.

[0026] Specifically, the table below shows values used in the encryption scheme between host 100 and slave 110, including example sizes for each value.

Data	Description	Size
SN	Slave serial number	4 bytes
EK	Secret encryption key	16 bytes
SEED	Slave random number seed	20 bytes
SID	Session identification	2 bytes
HRN	Host random number	8 bytes
SRN	Slave random number	8 bytes
SK	Session key	20 bytes

TABLE

Encryption Values

[0027] Each slave 110 stores in its memory 210 a unique slave serial number SN, a unique secret encryption key EK, a slave random number seed SEED and a session identifier SID in nonvolatile memory, such as memory 210. These values may be initially written to memory 210 as part of the manufacturing process for slave 110. Slave serial number SN is the unique serial number of slave 100. Secret encryption key EK is the secret key maintained in both slave 110 and host 100 that is used to derive the session key SK. Slave random number seed SEED is initialized with a true random number during the manufacturing process and updated by the slave 110 after each power cycle with a value derived from itself. The session identification SID is initialized to zero or some other value and is incremented or decremented by the slave 110 with each power cycle.

[0028] The operation of slave 110 will be described below with respect to Fig. 2.

[0029] Following slave 110 being reset, which may occur, for example, by slave 100 being initially connected to and powered by host 100, slave 110 calculates at 10 a new session identification SID based upon the then current session identification SID which is maintained in nonvolatile memory 210 within slave 110. The value of new session identification SID may be calculated, for example, by incrementing or decrementing the value of the current session identification SID.

[0030] In addition, following reset, slave 110 determines a new slave random number seed SEED at 20. According to the example embodiment, $SEED_0$ represents the true random number written to memory 210 for the slave random number seed SEED during the manufacturing process of the slave device 110. Slave random number seed $SEED_i$ is the value of slave random number seed SEED after the i-th subsequent power cycle. The i-th value of slave random number seed $SEED_i$ may be updated with the $SEED_{i-1}$ value of slave random number seed SEED following power up of slave device 110. In particular, $SEED_i$ may be computed using a secure algorithm, such as a secure hash algorithm (SHA). In this way, $SEED_i$ may be represented as:

$$SEED_i = SHA\text{-}1(SEED_{i-1})$$

Where "SHA-1" is the 160-bit secure hash function designed by the National Security Agency. It is understood that $SEED_i$ may be calculated using a different algorithm, including a different secure algorithm, such as a different SHA.

[0031] Slave random number seed $SEED_i$ is then used to compute at 30 a multi-byte random (or pseudorandom) number R, such as a 20 byte number, according to the equation:

$$R = SHA\text{-}1(SN \,\&\, SEED_i \,\&\, SID)$$

where "&" represents concatenation. The slave random number SRN for the session may be calculated to be a predetermined number of the most significant bytes of number R, such as the most significant eight bytes of R:

$$SRN = R[159:96]$$

[0032] It is understood that functions and algorithms other than SHA-1 may be utilized to generate SRN, such as another hash based algorithm.

[0033] The host 100 computes host random number HRN using a similar computation as described above for generating slave random number SRN, or any other random or pseudorandom number generator algorithm.

[0034] Host 100 and slave 110 communicate using commands and data that are encrypted. In an example embodiment, host 100 and slave 110 encrypt commands and data to be communicated with each other using a stream cipher. For example, host 100 and slave 110 may utilize the RC4 stream cipher due to its lower computational cost. It is understood, however, that any encryption scheme and/or stream cipher may be utilized by host 100 and slave 110 for communicating information therebetween. In general terms, an encryption session is established by exchanging values between host 100 and slave 110, from which host 100 and slave 110 independently calculate a session key SK based upon the exchanged values and a secret value known to each. The session key SK is then used to initialize the cipher, which as discussed in the example embodiment is a stream cipher.

[0035] To establish an encryption session at 40 for communicating encrypted information between host 100 and slave 110, host 100 sends the slave 110 host random value HRN. In response, the slave 110 sends host 100 the slave random number SRN and the session identification SID in response. Thereafter, both host 100 and slave 110 calculate at 50 the session key SK as follows:

$$SK = HMAC(EK, HRN \,\&\, SRN \,\&\, SID)$$

where HMAC is the hash-based message authentication code. As mentioned above, secret encryption key EK is known to both host 100 and slave 110, but is not transmitted on the bus 120. Session key SK may be, for example, 20 bytes in length and is not communicated over bus 120.

[0036] It is understood that other cryptographic functions, such as another hash-based function, may be utilized to generate session key SK. It is further understood that any encryption scheme could be used, and an example embodiment uses the RC4 stream cipher for its low computational cost.

[0037] According to an example embodiment, the most significant bytes of session key SK, such as SK[159:32] (16 bytes), may be used to initialize the stream cipher at 60 at the beginning of the encryption session. After initialization, the cipher produces a sequence of bytes K_0 K_1 K_2 K_3 .... Both host 100 and slave 110 compute the same K byte sequence because each initialized the cipher stream with the same session key SK. Host 100 then is able to encrypt at 60 a command packet for transmission to slave 110 by performing an exclusive OR operation ("XOR-ing") the command and data bytes with K_i, where the value i is incremented for each byte encrypted. Upon reception of the encrypted command packet, slave 110 then decrypts at 60 the packet received by XOR-ing the bytes with the same K bytes from the cipher. Similarly, the slave 110 encrypts at 60 the response packet and transmits the encrypted response packet which the host 100 is able to decrypt using the same K bytes used by slave 110 in encrypting the response packet.

[0038] As mentioned above, the most significant bytes of session key SK may be used for an encryption session. The least significant bytes of session key SK, in this case SK[31:0] (4 bytes), may be used to initialize at 70 the slave address generator by slave 110 and host 100.

[0039] Slave 110 may use a 10-bit address on bus 120. According to an example embodiment in which host 100 is a printing device and each slave device 110 is a different toner/ink cartridge, the most significant four bits of the slave address may be fixed and assigned a value corresponding to one type of ink or toner - cyan, magenta, yellow or black, for example. The least significant six bits of the 10-bit slave address may then be set by a pseudorandom number generator (PRNG) within slave 110 and host 100. After slave 110 is reset, the least significant six bits of its slave address, i.e., the slave's I^2C address, on bus 120 are 0. When host 100 instructs slave 110 to change its slave address at 70, the least significant six bits of the slave address are set from predetermined bits in the next value of the PRNG.

[0040] In accordance with an example embodiment, the PRNG may be a linear congruency generator (LCG) and may generate pseudorandom number X_n as follows:

$$X_n = 2891336453\ X_{n-1} + 1523469037 \bmod 2^{32}$$

where X_{n-1} represents the current value of pseudorandom number X_n. It is understood that other LCGs and/or PRNGs may be utilized for generating pseudorandom number X_n.

[0041] According to an example embodiment, the LCG is initialized with a predetermined number of bytes of session key SK, such as the least significant four bytes, SK[31:0], such that:

$$X_0 = SK[31:0]$$

After host 110 reads the response to the set address command, the next value of the LCG (X_n) is calculated and the slave (I^2C) address is set at 80 to be a predetermined a subset of bits of X_n. In an example embodiment,

$$\text{Slave Address}[5:0] = X_n\,[29:24]$$

Host 100 sends commands to change addresses to the slave 110 on a periodic basis, after which host 100 and slave 110 each compute the new address X_n for slave 110. Thereafter, slave 110 will not respond to address poll requests until after it has received a status request from host 100 using the new address X_n.

[0042] Host 100 and slave 110 communicate using command and response packets over bus 120. The packets contain a cyclic redundancy check (CRC) value to check for data errors in a packet. Data correction is accomplished by packet retransmission. If the CRC check fails in slave 110, then slave 110 returns a CRC response to the host 100. If the CRC check fails in host 100, then host 100 retransmits the previous command packet without advancing the stream cipher. In either case, host 100 retransmits the command packet again without changing its contents. This approach keeps host 100 and slave 110 synchronized in the cipher stream and also prevents the same cipher bytes from being used to encrypt different data.

[0043] The host-slave communication system described above uses an encrypted, packet-based communications scheme. A means for error detection and correction is provided utilizing CRC checks and packet retransmission. Host 100 and slave 110 exchange values so that each computes a session key SK from a secret key known to both

host 100 and slave 110 but not exchanged over bus 120. The session key SK is then used to initialize both the stream cipher and bus address function. With respect to the former, host 100 and slave 110 each encrypt and decrypt their communications by XOR-ing the transmitted/received data with bytes from the stream cipher. Host 100 periodically and/or occasionally changes slave addresses on the bus 120.

[0044] Advantages over existing systems include error detection and correction, encrypted communications, and a secure address change method that will always be synchronized between host 100 and slave 110. The error detection and correction increases reliability in noisy environments. The data encryption prevents an attacker from analyzing the bus traffic to learn the meaning of the commands and data shared between host 100 and slave 110. When implemented in a system in which host 100 is a printer and slave 110 is associated with a consumable toner or ink cartridge, the above-described address change method allows a slave 110 to be moved from printer to printer without issue while maintaining secure communication with the connected printer.

[0045] While the above describes example embodiments, many variations are possible within the scope of the present disclosure. For example, as discussed above a stream cipher is used to encrypt data because of its simplicity. Alternatively, a block cipher, such as the Advanced Encryption Standard (AES), would offer relatively greater security but at a higher computational cost. In such an alternative embodiment, some or all of the determined session key SK would be used in performing encryption and decryption on information to be transmitted and information received, respectively, in accordance with the particular block cipher utilized. The protocol corrects for errors by packet retransmission. Further, a forward error correction scheme could be used where error correction bits are included in the transmitted packet. Still further, a different addressed bus, such as the Universal Serial Bus (USB), could be used for bus 120 instead of a bus utilizing the I^2C protocol.

[0046] The foregoing description of one or more example embodiments has been presented for purposes of illustration. It is not intended to be exhaustive or to limit the application to the precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. It is understood that the invention

may be practiced in ways other than as specifically set forth herein without departing from the scope of the invention. It is intended that the scope of the application be defined by the claims appended hereto.

[0047] What is claimed is:

Claims:

1. A slave device, comprising:

a processor and memory coupled thereto having stored therein program code instructions which, when executed by the processor, cause the processor to:

following the slave device being reset, determine a seed value;

receive a host number from a host that is substantially random;

determine a session key based upon the determined seed value and the host number, the session key being substantially random; and

use the session key to perform encryption and decryption operations on data to be transmitted and data received by the slave device, respectively, and to determine an address value for the slave device for communicating with the host.

2. The slave device of claim 1, wherein the processor calculates a session identifier value following the slave device being reset, the session identifier being based upon a session identifier value prior to the slave device being reset, the session key being based upon the calculated session identifier value.

3. The slave device of claim 2, wherein the calculated session identifier value is the session identifier value prior to the slave device being reset that is incremented or decremented.

4. The slave device of claim 2, wherein the processor determines a substantially random number based upon the determined seed value, the calculated session identifier value and a serial number of the slave device, and wherein the session key is based upon the substantially random number.

5. The slave device of claim 1, wherein the determined seed value is determined using a secure hash algorithm.

6. The slave device of claim 1, wherein the determined seed value is substantially random.

7. The slave device of claim 1, wherein the session key is based upon a secret encryption key of the slave device.

8. The slave device of claim 1, wherein a hash-based message authentication code (HMAC) is used to determine the session key.

9. The slave device of claim 1, wherein a first portion of the session key is used by the processor for performing the encryption and decryption operations on information to be transmitted and information received by the slave device, respectively, and a second portion of the session key is used for determining the address value for the slave device for communicating with the host.

10. The slave device of claim 1, wherein the encryption and decryption operations form part of a stream cipher for communicating with the host.

11. The slave device of claim 1, wherein the determined seed value is based upon a seed value of the slave device prior to the slave device being reset.

12. A computer program product stored in a non-transitory storage medium and having instructions which when executed by a processor in a slave device causes the processor to:

following the slave device being reset, determine a seed value;

receive a host number from a host that is substantially random;

determine a session key based upon the determined seed value and the host number, the session key being substantially random; and

use the session key to perform encryption and decryption operations on data to be transmitted and data received by the slave device, respectively, and determining a new address value for the slave device for communicating with the host.

13. The computer program product of claim 12, wherein the instructions cause the processor to calculate a session identifier value following the slave device being reset, the session identifier value being based upon a session identifier value prior to the slave

device being reset, the session key being based upon the calculated session identifier value.

14. The computer program product of claim 13, wherein the instructions cause the processor to determine a substantially random number based upon the determined seed value, the calculated session identifier value and a serial number of the slave device, and wherein the session key is based upon the substantially random number.

15. The computer program product of claim 12, wherein the determined seed value is determined using a hash algorithm.

16. The computer program product of claim 12, wherein the determined seed value is substantially random.

17. The computer program product of claim 12, wherein the session key is based upon a secret encryption key of the slave device.

18. The computer program product of claim 12, wherein a first portion of the session key is used for performing the encryption and decryption operations and a second portion of the session key is used for determining the new address value for the slave device for communicating with the host.

19. The computer program product of claim 12, wherein a hashed based message authentication code (HMAC) is used to determine the session key.

20. The computer program product of claim 12, wherein the encryption and decryption operations comprise part of an RC4 stream cipher for communicating with the host.

21. The computer program product of claim 12, wherein the determined seed value is based upon a seed value of the slave device prior to the slave device being reset.

22. A slave device, comprising:

a processor and memory coupled thereto configured to:

following the slave device being reset, determine a seed value;

receive a host number from a host that is substantially random;

determine a session key based upon the determined seed value and the host number, the session key being substantially random; and

use the session key to perform to determine a new address value for the slave device for communicating with the host.

23. The slave device of claim 22, wherein the processor and memory are further configured to use the session key to perform encryption and decryption operations on data to be transmitted and data received by the slave device, respectively.

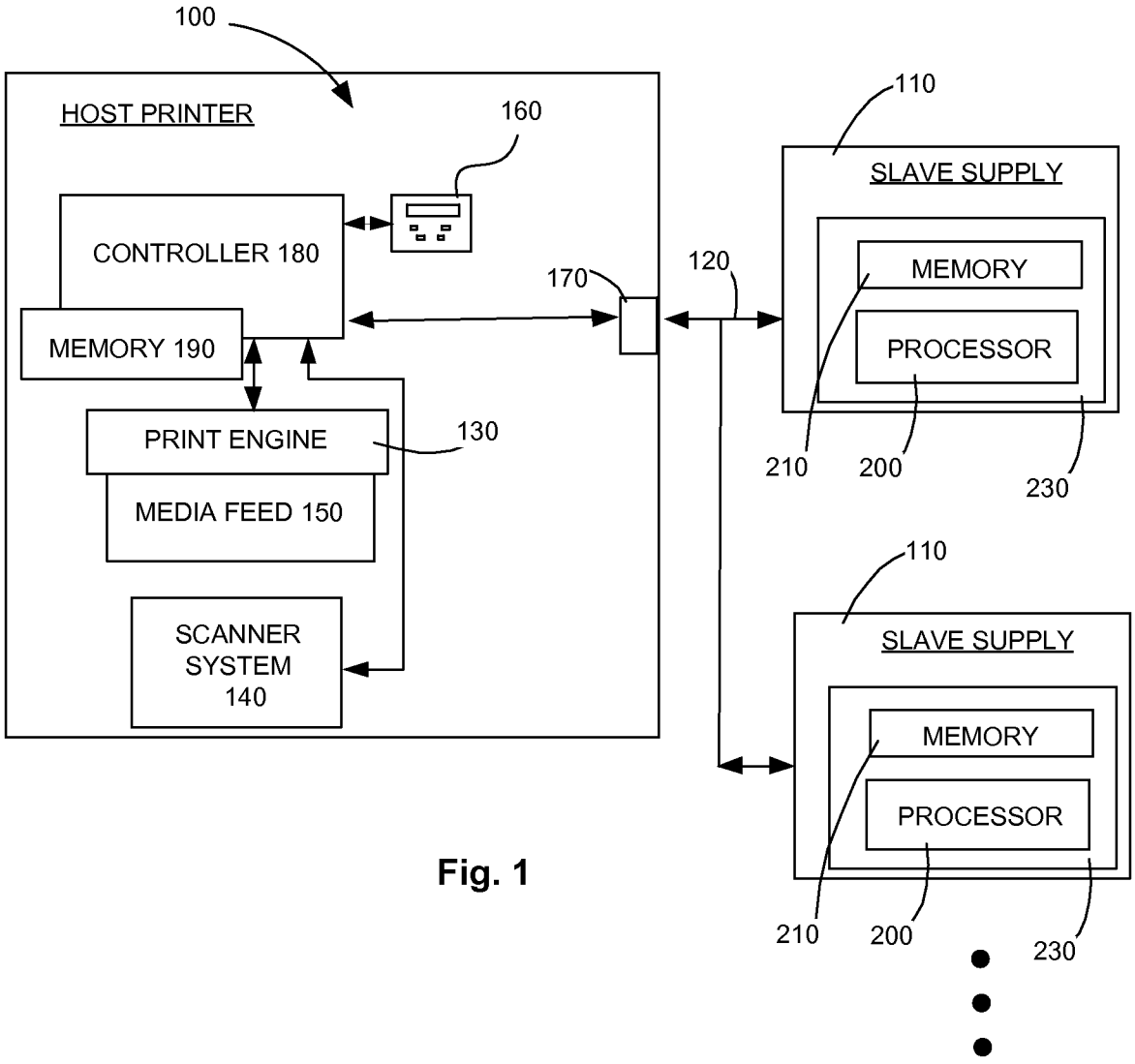
100
HOST PRINTER
160
CONTROLLER 180
MEMORY 190
170
120
PRINT ENGINE
130
MEDIA FEED 150
SCANNER SYSTEM 140
110
SLAVE SUPPLY
MEMORY
PROCESSOR
210
200
230
110
SLAVE SUPPLY
MEMORY
PROCESSOR
210
200
230

Fig. 1

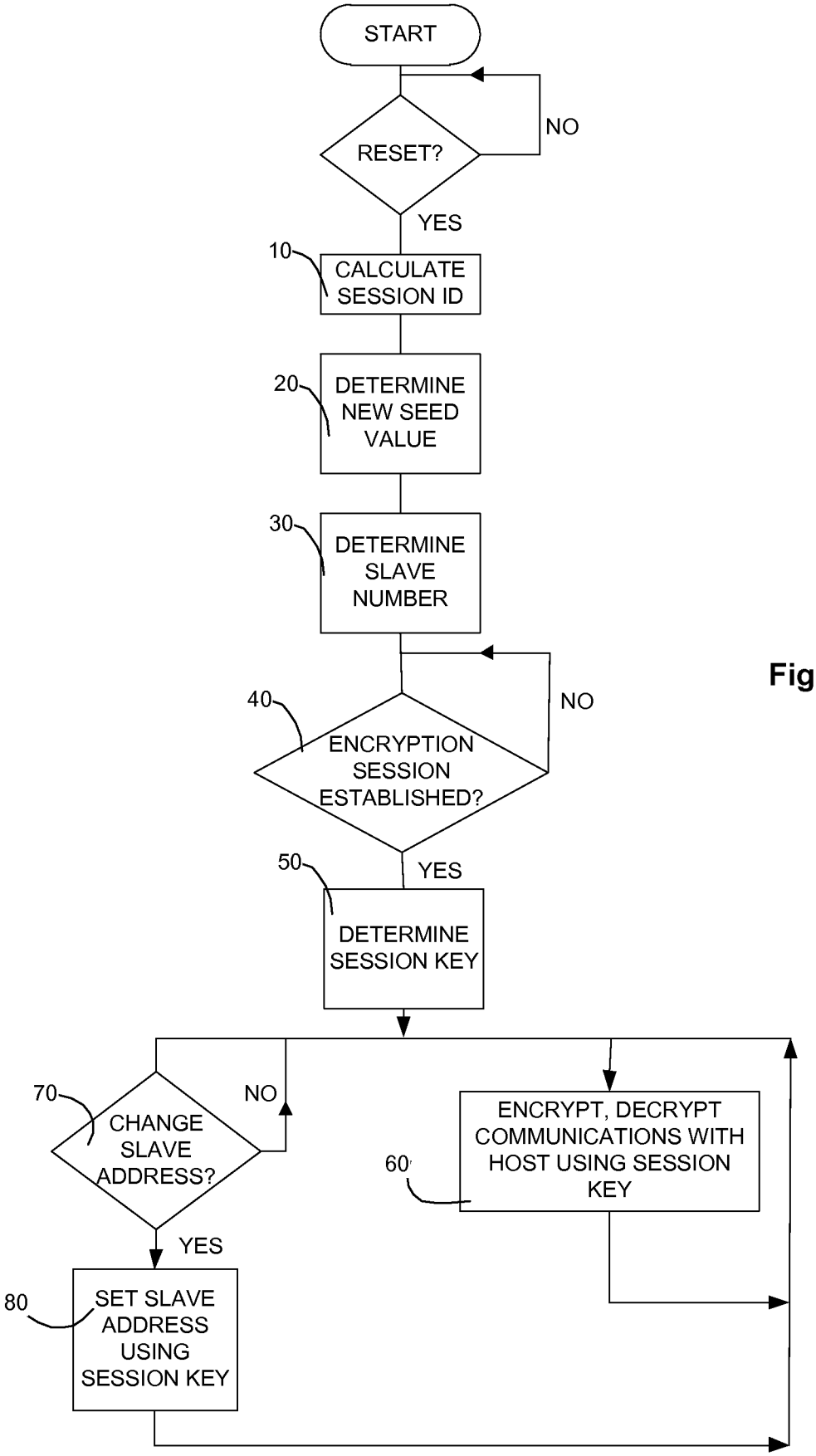
START
RESET?
NO
YES
10
CALCULATE SESSION ID
20
DETERMINE NEW SEED VALUE
30
DETERMINE SLAVE NUMBER
40
ENCRYPTION SESSION ESTABLISHED?
NO
YES
50
DETERMINE SESSION KEY
70
CHANGE SLAVE ADDRESS?
NO
YES
80
SET SLAVE ADDRESS USING SESSION KEY
60
ENCRYPT, DECRYPT COMMUNICATIONS WITH HOST USING SESSION KEY

Fig. 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 12/54108

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 15/16 (2012.01)
USPC - 709/210
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC: 709/210; IPC(8): G06F 15/16 (2012.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 455/403,410,411; 712/31; 726/2,27; 709/208-211

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Patbase (Full-text: AU BE BR CA CH CN DE DK EP ES FI FR GB IN JP KR SE TH TW US WO); Google Scholar
Search Terms: slave, ink cartridge, toner cartridge, developer unit, fuser unit, print engine, processor, CPU, memory, RAM, ROM, storage, slave reset, seed, host, master, random, pseudo-random, session key, encryption, decryption, transmit, communic

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 2008/0184341 A1 (SEBESTA et al.) 31 July 2008 (31.07.2008), abstract and para [0037]-[0039], [0041], [0044]-[0045], [0048], [0051]-[0056], [0061]-[0062], [0067].	1-2, 4-9, 11-19, 21-23 --- 3, 10, 20
Y	US 2010/0306431 A1 (ADKINS et al.) 02 December 2010 (02.12.2010), abstract and para [0031]-[0032], [0034], [0040]-[0041], [0047].	3
Y	US 2011/0074850 A1 (WALMSLEY et al.) 31 March 2011 (31.03.2011), abstract and para [1199]-[1202], [4352].	10, 20
A	US 2008/0259711 A1 (SHIPTON et al.) 23 October 2008 (23.10.2008), entire document.	1-23

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 October 2012 (23.10.2012)	27 NOV 2012
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774