(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 473 615 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
06.04.2011 Bulletin 2011/14

(51) Int Cl.:
G06F 21/04 *(2006.01)*

(21) Application number: **04010225.3**

(22) Date of filing: **29.04.2004**

(54) **Secure communication with a keyboard**

Verfahren zum sicheren Kommunizieren mit einer Tastatur

Procédé de communication sécurisée avec un clavier

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PL PT RO SE SI SK TR

(30) Priority: **02.05.2003 US 428675**

(43) Date of publication of application:
03.11.2004 Bulletin 2004/45

(73) Proprietor: **MICROSOFT CORPORATION**
Redmond, WA 98052 (US)

(72) Inventors:
- **Peinado, Marcus**
  Bellevue
  Washington 98008 (US)
- **Benaloh, Josh**
  Redmond
  Washington 98052 (US)

(74) Representative: **Grünecker, Kinkeldey, Stockmair & Schwanhäusser**
Anwaltssozietät
Leopoldstrasse 4
80802 München (DE)

(56) References cited:
EP-A- 1 286 242

- **G. TREAT: "Keyboard encryption" IEEE POTENTIALS, [Online] vol. 21, no. 3, August 2002 (2002-08), - September 2002 (2002-09) pages 40-42, XP002403208 ISSN: 0278-6648 Retrieved from the Internet: URL:http:// ieeexplore.ieee.org/iel5/45/221 95/01033666.pdf? arnumber=1033666> [retrieved on 2006-10-13]**

Description

FIELD OF THE INVENTION

[0001] The present invention relates generally to the field of computer security. More particularly, the invention relates to the secure use of a keyboard over a communication channel that may be subject to interception or other types of tampering.

BACKGROUND OF THE INVENTION

[0002] A keyboard communicates user-entered data to an electronic device, such as a computer. When a user presses a key on the keyboard, the keyboard generates data representative of the particular key that was pressed (e.g., the ASCII code for the letter "e"), and this data is received by a component in the computer, such as a device driver. The device driver then presents the data to whatever program running on the computer is currently receiving input (e.g., by placing the data into the input buffer for whichever application program is active).

[0003] One problem that arises in using a keyboard to receive data is when the data is sensitive, or otherwise needs to be kept secret. For example, a secure application (or a secure service of an operating system) may ask the user to enter a password, which should not be generally divulged to the public at large. However, the path leading from the keyboard to the software component that will receive the data is not secure, since there are several opportunities to intercept the data. For example, the data will often travel on a bus that is subject to snooping, and will be handled by a device driver that may be subject to tampering (or that the operating system will allow to be replaced with a non-secure device driver that stores and divulges the information that the driver handles). In other words, there are several opportunities to observe or tamper with secret data on its way from the keyboard to its ultimate destination.

[0004] In general, it is possible to encrypt data for transmission between two components that are connected by a non -secure channel. However, many encryption techniques cannot easily be applied in the context of a keyboard, due to various factors, such as key management issues, the possibility of replay attacks, and the fact that the relatively small range of data that can be generated by a keyboard would make an ordinary cipher on keyboard communications relatively easy to break if a moderately-sized sample of ciphertext can be intercepted.

[0005] In view of the foregoing, there is a need for a technique that facilitates secure communication with a keyboard.

[0006] EP-A-1 286 242 discloses a scrambling device interconntected between an input device and an input device decoder. The scrambling devices can be controlled to operate in a normal mode in which data passes unchanged and a secure mode in which data is encrypted. An uses case disclosed by the document related to the input of an Personal Identification Number (PIN).

[0007] G. Treat: "Keyboard encryption" IEEE Potentials, vol. 21, no. 3, August 2002 - September 2002, pages 40 to 42 relates to encrypting keyboard data wither by using a plug-in device between the keyboard and the computer or using a modified microcontroller in the keyboard and decryption software loaded onto an existing BIOS chip of the computer.

SUMMARY OF THE INVENTION

[0008] It is the object of the present invention to avoid replay attacks in an encrypted communication channel between a keyboard and a software component of a computer.

[0009] This object is solved by the subject matter of the independent claims.

[0010] Preferred embodiments are defined by the dependent claims..

[0011] The present invention provides a technique for secure communication between two components through a non-secure communication channel. The technique uses an encryption scheme that is particularly well-adapted for a keyboard, and that addresses problems that would exist in applying a standard encryption scheme to a keyboard.

[0012] A keyboard in accordance with the invention stores a key and a constant value that is used for initialization of the encryption scheme. A component (e.g., an application running on a computer) stores the same key and the same constant value that are stored at the keyboard. In order to initiate a secure session between the component and the keyboard each generates a nonce, and then exchanges nonce with the other, so that the keyboard and the component are each in possession of both nonces. The keyboard and the component then compute two initial values, each of which is based on the two nonces, the key, and the constant value. For example, the first initial value may be created by using the CBC-3DESMAC algorithm, where CBC-3DESMAC uses the stored constant value as its initial chaining value and applies the key to a message created based on the two nonces. (CBC-3DESMAC refers to applying triple encryption according to the Data Encryption Standard (DES) algorithm with cipher block chaining, and using the final ciphertext block to create a Message Authentication Code (MAC)). Preferably the second initial value is created by inverting the bits in the first initial value (i.e., perform an "exclusive or" operation between the first initial value and the number Oxffffffffffffffff). Since the keyboard and the component compute the first and second initial values in the same way, they are both in possession of the same two initial values.

[0013] In an alternative preferred embodiment, the keyboard and the component are equipped with two constant values, and the first and second initial values can be created by applying CBC-3DESMAC to the message that is based on both nonces, using the first constant to create the first initial value, and the second constant to create the second initial value.

[0014] After the first and second initial values have been created, the keyboard is ready to communicate encrypted data, and the component that will receive the data is ready to decrypt and verify the data. When data is entered into the keyboard, the keyboard encrypts the data based on the first initial value and the key. Preferably, the keyboard encrypts the data with the above-mentioned key using CBC-3DES (triple-DES with cipher block chaining), with the first initial value being used to prime the cipher block chain. The keyboard also preferably creates a MAC for each unit of data using CBC-3DESMAC, where CBC-3DESMAC applies the above-mentioned key and uses the second initial value to prime the cipher block chain. Preferably, each keystroke is encrypted in a separate encryption block, and the entire stream of data generated at the keyboard during a session constitutes a chain of cipher blocks, since this technique allows the same keystroke (e.g., the letter "e") to appear as different ciphertext depending upon the keystroke that preceded it.

[0015] Once the encrypted data and MAC(s) have been received at the receiving component, the receiving component uses the above-mentioned key and the first and second initial values to decrypt and verify the received data.

[0016] Other features of the invention are described below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] The foregoing summary, as well as the following detailed description of preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings exemplary constructions of the invention; however, the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

[0018] FIG. 1 is a block diagram of an exemplary computing environment in which aspects of the invention may be implemented;

[0019] FIG. 2 is a block diagram of a first exemplary environment in which communication between a keyboard and a component may take place over a non-secure channel;

[0020] FIG. 3 is a block diagram of a second exemplary environment in which communication between a keyboard and a component may take place over a non-secure channel;

[0021] FIG. 4 is a block diagram of a keyboard and a component that have been configured for secure communication, and which exchange nonces, in accordance with aspects of the invention;

[0022] FIG. 5 is a flow diagram of a process for engaging in a secure communication session between a keyboard and a component; and

[0023] FIG. 6 is a block diagram of a first exemplary environment in which keyboards and components may be distributed to engage in secure communication according to aspects of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Exemplary Computing Arrangement

[0024] FIG. 1 shows an exemplary computing environment in which aspects of the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

[0025] The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, embedded systems, distributed computing environments that include any of the above systems or devices, and the like.

[0026] The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network or other data transmission medium. In a distributed computing environment, program modules and other data may be located in both local and remote computer storage media including memory storage devices.

[0027] With reference to FIG. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus (also known as Mezzanine bus).). The system bus 121 may also be implemented as a point-to-point connection, switching fabric, or the like, among the communicating devices.

[0028] Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CDROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

[0029] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, FIG. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[0030] The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, FIG. 1 illustrates a hard disk drive 140 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156, such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through an non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0031] The drives and their associated computer storage media discussed above and illustrated in FIG. 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 190.

[0032] The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0033] When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Security of Communication Between a Keyboard and a Component

[0034] The invention addresses the problem of how a keyboard can be used to communicate securely with a component that requires input from the keyboard. FIG. 2 shows an exemplary scenario of such communication. In FIG. 2, keyboard 162 communicates with component 204. Component 204 can be any type of component - e.g., a program that is executing on a computer, a piece of hardware, etc. Communication from keyboard 162 to component 202 passes through a communication channel that includes at least some non-secure portion 204. That is, as the data that represents keystrokes passes through some channel on its way from keyboard 162 to component 202, there may be some opportunity for a third party to intercept or tamper with the data. This interception or tampering may be a problem if, for example, the information that is being typed at keyboard 162 is a secret password that should not be revealed to the general public.

[0035] FIG. 3 shows a particular scenario in which secure communication between a keyboard and a component is desired. In FIG. 3, keyboard 162 is used to provide input to software that is running on computer 110. In the example of FIG. 3, keyboard 162 is a keyboard adapted for use with a Universal Serial Bus (USB) 302. (For brevity, such a keyboard shall be referred to as a USB keyboard.) Keyboard 162 receives keystrokes, and places bytes representative of those keystrokes onto USB 302, where the bytes are picked up by USB driver 304. Driver 304 then communicates those bytes to their ultimate destination, which, in the example of FIG. 3, is software 306. Software 306 is an example of component 202 (shown in FIG. 2).

[0036] In the example of FIG. 3, there are two operating systems 134(1) and 134(2) running on computer 110. Operating system 134(1) is a typical operating system, such as MICROSOFT WINDOWS XP, Unix, Linux, Solaris, etc. Operating system 134(2) is a "high-assurance" operating system that is used for trusted applications. For example, operating system 134(2) may be associated with a "curtained" memory that is not accessible outside of operating system 134(2), and operating system 134(2) may store secret information (e.g., cryptographic keys, passwords, etc.) in that curtained memory, so that only certain special trusted applications that are permitted to execute under operating system 134(2) are able to read that secret information. Operating system 134(2) is "high assurance" in the sense that the public is entitled to a very high level of assurance that it will perform its function correctly - i.e., if protecting secret information is one of the intended functions of operating system 134(2), the public is entitled to a very high level of assurance that operating system 134(2) will not divulge that secret information. Part of being able protect secret information may include being able to receive typed secrets (e.g., passwords) without divulging these secrets to the outside world. Operating system 134(2) may not trust driver 304 to handle such secret information, since driver 304 is under the control of operating system 134(1) (and operating system 134(1) might allow a hacker to read information directly from USB 302, or substitute a nefarious driver that would store and reveal the secret information). Thus, operating system 134(2) needs a way to receive information from keyboard 162 through operating system 134(1) without concern that the secret information will be divulged by acts arising in operating system 134(1).

[0037] It should be understand that while the example of FIG. 3 shows keyboard 162 as communicating with computer 110 through Universal Serial Bus 302, the scenarios described above apply regardless of the exact means by which keyboard 162 communicates with computer 110, and thus the invention is not limited to USB keyboards.

[0038] FIG. 4 shows how keyboard 162 and component 202 may be configured to participate in secure communication through a non-secure channel. Keyboard 162 and component 202 each store a copy of cryptographic key 402. Keyboard 162 and component 202 also preferably store a constant value 404, which is used as the initial value for a particular preferred cryptographic technique, as more particularly described below. In a further preferred embodiment, keyboard 162 and component 202 may store (in addition to the key) two constant values instead of one; these two constant values may be used in a cryptographic technique as described below. Keyboard 162 may, for example, contain an onboard non-volatile semiconductor that stores key 402 and constant 404, or may have a port that receives a removable storage medium on which key 402 and constant 404 are stored. In the case where component 202 is a software component, key 402 and constant 404 may be stored in component 202's data space. It will be understood, however, that the invention is not limited to any particular manner of storing key 402 and 404.

[0039] At the outset of secure communication between keyboard 162 and component 202, keyboard 162 and component 202 may generate and exchange nonces. That is, keyboard 162 generates nonce 412 and sends nonce 412 to component 202. Component 202 generates nonce 414 and sense nonce 414 to keyboard 162. As is known in the art, a nonce is a piece of data that is used in cryptographic applications - often to authenticate an entity cryptographically, or to prime an encryption session with a not-easily-reproduced element on which the encryption can be made dependent. Nonces 412 and 414 may be used to create initial values for encryption and authentication of data transmitted between keyboard 162 and component 202, as more particularly described below.

<u>Process of Securely Sending Data from a Keyboard to a Component</u>

[0040] FIG. 5 shows a process by which keyboard 162 and component 202 may engage in a session wherein component 202 securely receives data from keyboard 162. The process of FIG. 5 provides for both encryption (which protects against interception of the transmitted data), and authentication (which protects against modification of the transmitted data). However, it will be understood that either encryption or authentication alone can be used, depending on the security requirements of the transmission. For example, if modification of the data can be tolerated but interception cannot be tolerated, then encryption alone can be used. Conversely, if interception of the data can be tolerated, but modification of the data cannot be tolerated, then authentication alone can be used.

[0041] Initially, keyboard 162 and component 202 exchange 502 nonces . For example, as described above in connection with FIG. 4, keyboard 162 may generate nonce 412 and send it to component 202, and component 202 may generate nonce 414 and sent it to keyboard 162. Techniques for generating nonces are known in the art, and thus are not described at length herein. As some examples, nonces 412 and 414 could be generated based on a random number, the contents of some region of memory, time, temperature, phase of the moon, etc., or any other factor that is likely to change often and has a sufficient range that it is unlikely that either keyboard 162 or component 202 will produce the same nonce twice.

[0042] After nonces 412 and 414 are exchanged 502, keyboard 162 and component 202 are each in possession of both nonces. Keyboard 162 and component 202 then use a commonly agreed upon formula to compute 504 two initial values - IV_c and IV_m - as functions of both nonces and key 402. That is, if K = key 402, N_1 = nonce 412, and N_2 = nonce 414, then

$$IV\_c = f(K, N_1, N_2);$$

and

$$IV\_m = g(K, N_1, N_2).$$

The functions f and g can be any functions. In a preferred embodiment,

$$f(K, N_1, N_2) = CBC\text{-}3DESMAC_K(const\_IV, N_1 \mid N_2);$$

and

$$g(K, N_1, N_2) = f(K, N_1, N_2) \text{ xor } 0xffffffffffffffff,$$

where const_IV is equal to constant value 404 (shown in FIG. 4). In a further preferred embodiment, where the keyboard and the component share two constant values (e.g., const_IV_1 and const_IV_2), the functions f and g can alternatively be computed as follows:

$$f(K, N_1, N_2) = CBC\text{-}3DESMAC_K(const\_IV\_1, N_1 \mid N_2);$$

and

$$g(K, N_1, N_2) = \text{CBC-3DESMAC}_K(\text{const\_IV\_2}, N_1 \mid N_2),$$

(The operator "|" means concatenation, so that $N_1| N_2$ is the value resulting from concatenating N_1 with N_2. "xor" is the bitwise "exclusive or" operation, such that A xor B is the value resulting from setting to "1" any bit that is a "1" in either A or B but not both, and setting all other bits to zero.) CBC-3DESMAC$_K$(-const IV, $N_1| N_2$) is a cryptographic function, whose meaning is known in the art and described in greater detail below.

[0043] After IV_c and IV_m have been computed, communication between keyboard 162 and component 202 can begin. Keyboard 162 receives a keystroke - i.e., by an operator pressing one of the keys (or certain combinations of keys, such as <SHIFT> and "A", or <CTRL> and "A") (step 506). The keyboard next encrypts 508 the keystroke; the encryption is preferably based on key 402 and IV_c . In a preferred embodiment, the keystrokes are encrypted using CBC-3DES, with key 402 as the key and IV_c as the initial value. CBC-3DES is a cryptographic algorithm that is known in the art and described in greater detail below below. Additionally, keyboard 162 computes 510 a message authentication code (MAC) for the keystroke, preferably based on key 402 and IV_m. In a preferred embodiment, the message authentication code is created using CBC-3DESMAC, with key 402 as the key and IV_m as the initial value. As noted above, CBC-3DESMAC is known in the art and described in greater detail below.

[0044] After the keyboard has created both the encrypted keystroke data and the MAC, component 202 receives 512 the encrypted keystroke data and MAC from keyboard 162 (step 512). Component 202 then decrypts 514 the data using key 402 and IV_c, and also verifies the data using key 402 and IV_m (step 514). The process then returns to step 506 to receive the next entry at the keyboard.

<u>The Cryptographic Functions CBC-3DES and CBC-3DESMAC</u>

[0045] CBC-3DES is a cryptographic function that combines the data encryption standard (DES) with cipher block chaining (CBC). "3DES" means that the DES encryption algorithm is applied to a given block of data three times ("triple-DES"). DES encrypts data by applying a key to the data in a known manner. DES encrypts a long message by dividing the message into smaller blocks, and encrypting the individual blocks. (When "triple-DES" is used, the DES algorithm is applied to each block three times in order to produce the ciphertext for that block.) DES (and triple-DES) can encrypt each block of data using just a key; however, when cipher block chaining is used, the encryption of one block is based not only on the key, but also on the ciphertext that was produced by encrypting the last block. Thus, encryption of a given block is based on two inputs: the key, and the ciphertext that resulted from encrypting the previous block. Since the first block of data to be encrypted has no "previous" block, the cipher block chaining process must be primed with an "initial value" - that is, the first block of data is encrypted based on the key and some initial value. The initial value is not used in the encryption of subsequent blocks, but may indirectly influence how those blocks are encrypted (since the first block's ciphertext is based on the initial value, the second block's ciphertext is based on the first block's ciphertext, and so on).

[0046] In view of the preceding discussion, the phrase "CBC-3DES$_K$(IV, message)," means encrypting "message" with the key K, using triple-DES and cipher block chaining, where IV is the initial value for the cipher block chain.

[0047] CBC-3DESMAC is a way of using CBC-3DES to produce a message authentication code (MAC). In particular, the phrase CBC-3DESMAC$_K$(IV, message) means that "message" is encrypted with a key K using triple-DES and cipherblock chaining, and using IV as the initial value for the cipher block chain. However, since the goal of CBC-3DESMAC is only to produce a MAC for the message instead of a complex ciphertext for the message, only the last block of ciphertext is saved, and the remaining blocks of ciphertext may be discarded. This last block of ciphertext may be used as a MAC, since - even given a constant key and a constant IV - different messages are unlikely to produce the same final block (or, more precisely, if each block can represent 2" different values, there is only a 1 in 2" chance that any two messages will have the same final block).

[0048] It should be noted that the particular choice of CBC-3DES, as well as the way in which it is used, particularly advantageous for encrypted keyboard communication. Since the domain of messages to be encrypted is small (e.g., on the order of 128 different ASCII characters), cipher block chaining is particularly useful in keeping the cipher from being broken. If straight encryption were used (without chaining), then, within a given session, each character would encrypt to the same ciphertext each time it was typed - e.g., typing an "e" would always produce the same ciphertext. By making an educated guess (e.g., by using the fact that "e" is the most commonly occurring letter in the English language), one could more easily break such a cipher. Chaining all of the input in a session makes the cipher harder to break by ensuring that the same data may appears as different ciphertext depending upon where it appears in the input stream (e.g., an "e" may not always produce the same ciphertext). Additionally, changing the encryption for each session by creating a new initial value based on nonces prevents observers from detecting patterns of usage that they could use to compromise security (e.g., if the first text typed in every session is the password, an observer could capture the ciphertext for the password and institute a replay attack). Moreover, the size of cipher blocks used by DES is particularly well suited, since DES operates on 8-byte blocks, and most keyboard protocols transmit data in blocks that can fit into this size (E.g., the USB standard also deals in 8-byte blocks, so each USB block can fit into one DES block with no wasted space.) However, it should be understood that any other block cipher could be used, and chaining concepts similar to CBC could be applied to such a block cipher.

[0049] It should further be noted that, for the same reasons that the encryption scheme described herein is particularly well-suited to a keyboard, that encryption scheme is also well suited to certain other types of input devices, such as a mouse (or other pointing device). These input devices share various features in common with a keyboard, such as a small vocabulary, and a limited ability to execute a complicated encryption algorithm.

<u>Exemplary Use of Keyboard that Encrypts Data</u>

[0050] FIG. 6 shows an exemplary environment in which a keyboard that performs encryption may be used with components that require secure communication. In the example of FIG. 6 manufacturer 602 manufactures a plurality of keyboards 162(1), 162(2), ..., 162(n), and distributes these keyboard for public use. Each of the keyboard 162(1), 162 (2), ..., 162(n) incorporates key 402 and constant value 404 (shown in FIG. 4) (or incorporates some means by which key 402 and constant value 404 can be accessed externally, such as by means of a port for a removable semiconductor memory). Manufacturer 604 produces components 202(1), 202(2), ..., 202(m) that benefit from securely communicating with a keyboard. Each of components 202(1), 202(2), ..., 202(n) incorporates key 402 and constant value 404 (or is somehow able to receive the key and constant value). Components 202(1), 202(2), ..., 202(m) may now receive input from keyboards 162(1), 162(2), ..., 162(n), through the techniques described above.

[0051] Manufacturer 602 may have a preexisting relationship with manufacturer 604, so that both manufacturers can agree on a key 402 and a constant 404 that should be incorporated for secure communication. In one example manufacturers 602 and 604 are the same entity. In another example, manufacturer 604 is a manufacturer of components 202 (1), 202(2), ..., 202(m), who would like those components to be able to receive data from secure keyboards, and manufacturer 602 is a manufacturer of keyboards, whom manufacturer 604 has deemed sufficiently trustworthy to manufacture keyboards for secure communication with components 202(1), 202(2), ..., 202(m), and to hold key 402 and/or constant 404.

[0052] It is noted that the foregoing examples have been provided merely for the purpose of explanation and are in no way to be construed as limiting of the present invention. While the invention has been described with reference to various embodiments, it is understood that the words which have been used herein are words of description and illustration, rather than words of limitations. Further, although the invention has been described herein with reference to particular means, materials and embodiments, the invention is not intended to be limited to the particulars disclosed herein; rather, the invention extends to all functionally equivalent structures, methods and uses, such as are within the scope of the appended claims. Those skilled in the art, having the benefit of the teachings of this specification, may effect numerous modifications thereto and changes may be made without departing from the scope and spirit of the invention in its aspects.

Claims

1. A method of communicating with a keyboard (162) comprising:

receiving, at a component, a first nonce from the keyboard;
sending from the component (202) a second nonce (412) to the keyboard;
creating a first initial value and a second initial value by applying triple-DES and cipher block chaining to a combination of said first nonce and said second nonce, using a key and a third initial value that is known both to the keyboard and to the component;
receiving, at a component, a plurality of data that have been encrypted with triple-DES and cipher block chaining using the key and the first initial value from the keyboard, each separate keystroke received from said keyboard being included within a separate one of said plurality of data, each one of the plurality of data being encrypted using a separate block of said triple-DES and cipher block chaining;
decrypting the plurality of data based on said first initial value and said key.

2. The method of claim 1, further comprising:

receiving at the component (202), a plurality of message authentication codes corresponding to the plurality of data from the keyboard (162), said message authentication codes having been created with triple-DES and cipher block chaining using said key and the second initial value different from said first initial value, said second initial value being known both to the component and to the keyboard;
verifying the plurality of data by using the plurality of message authentication codes.

3. The method of claim 2, wherein creating a first initial value and a second initial value further comprising:

creating the first initial value by applying triple-DES and cipher block chaining to a combination of the first nonce and the second nonce, using the key and a third initial value that is known both to the keyboard and to the component; and
creating the second initial value by applying triple-DES and cipher block chaining to a combination of the first nonce and the second nonce, using the key and a fourth initial value that is known both to the keyboard and to the component.

4. The method of claim 1, wherein the plurality of data are received through a channel whose behavioral integrity is not trusted by the component.

5. The method of claim 4, wherein the component (202) comprises a first operating system which executes on a computing device along with a second operating system, the first operating system distrusting, in at least some respect, the behavior of the second operating system, the keyboard communicating with the first operating system through a driver controlled by the second operating system.

6. The method of claim 1, wherein the keyboard comprises a USB keyboard.

7. The method of claim 6, wherein said triple-DES and cipher block chaining encrypt data in blocks having a predetermined size, and wherein the keyboard communicates data in blocks of said predetermined size.

8. The method of claim 1, wherein each of the plurality of data is generated based on an individual keystroke received at the keyboard.

9. A computer-readable medium encoded with computer-executable instructions to perform a method of securely receiving input at a software component (202) from a keyboard (162), the method comprising:

receiving, at a software component, a first nonce (412) from the keyboard;
sending from the software component a second nonce (414) to the keyboard;
creating a first initial value and a second initial value by applying triple-DES and cipher block chaining to a combination of said first nonce and said second nonce, using a key and a third initial value that is known both to the keyboard and to the component;
receiving, at the software component, a plurality of encrypted keystrokes from the keyboard, the encrypted keystrokes having been created at the keyboard by encrypting input keystrokes received at the keyboard with triple-DES and cipher block chaining using the key and the first initial value, each separate keystroke received from said keyboard being included within a separate one of said plurality of data, each one of the plurality of data being encrypted using a separate block of said triple-DES and cipher block chaining; and
at the software component, decrypting the plurality of encrypted keystrokes using the key and the first initial value.

10. The computer-readable medium of claim 9, wherein the software component comprises (202) a first operating system which executes on a computing device along with a second operating system, the first operating system distrusting, in at least some respect, the behavior of the second operating system, the keyboard communicating with the first operating system through a driver controlled by the second operating system.

11. The computer-readable medium of claim 9, wherein the method further comprises:

receiving at the software component (202), a plurality of message authentication codes corresponding to the plurality of encrypted keystrokes from the keyboard (162), said message authentication codes having been created with triple-DES and cipher block chaining using said key and the second initial value different from said first initial value, said second initial value being known both to the software component and to the keyboard;
verifying the plurality of encrypted keystrokes by using the plurality of message authentication codes.

12. The computer-readable medium of claim 11, wherein creating a first initial value and a second initial value further comprises:

creating the first initial value by applying triple-DES and cipher block chaining to a combination of the first nonce (412) and the second nonce (414), using the key and a third initial value that is known both to the keyboard and to the software component (202); and
creating the second initial value applying triple-DES and cipher block chaining to a combination of the first nonce and the second nonce, using the key and a fourth initial value that is known both to the keyboard and to the software component.

13. The computer-readable medium of claim 9, wherein said triple-DES and cipher block chaining encrypt data in blocks having a predetermined size, and wherein the keyboard communicates data in blocks of said predetermined size.

14. A keyboard (162) comprising:

one or more storage locations that store a first initial value and a key;
an encryption component (202) that is adapted to receive a first nonce (412) from the recipient through the communication interface, to send a second nonce (414) to the recipient through the communication interface, and to create the first initial value by applying triple-DES and cipher block chaining to a combination of the first nonce and the second nonce, using the key and a second initial value that is known both to the keyboard and to the component, wherein said encryption component encrypts input data received at the keyboard with triple-DES and cipher block chaining using said key and said first initial value, whereby encrypted data is created based on said input data, each individual one of said input data being representative of a separate keystroke received at said keyboard, each of said individual ones of said input data being encrypted using a separate block of said triple-DES and cipher block chaining; and
a communication interface that communicates said encrypted data to a device external to the keyboard, said encrypted data being destined for a recipient that knows said first initial value and said key.

15. The keyboard (162) of claim 14, wherein the encryption component further creates a plurality of message authentication codes corresponding to the encrypted data or the input data, said message authentication codes having been created with triple-DES and cipher block chaining using said key and an initial value different from said first initial value.

16. The keyboard (162) of claim 14, wherein the keyboard comprises a USB keyboard.

17. The keyboard (162) of claim 14, wherein the communication interface communicates said encrypted data to a driver that is controlled by a first operating system that executes on said device whereupon said encrypted data is communicated to said recipient, said recipient being a second operating system that executes on said device or a program that executes under said second operating system, said second operating system distrusting, in at least some respect, the behavior of the first operating system.

Patentansprüche

1. Verfahren zum Kommunizieren mit einer Tastatur (162), das umfasst:

Empfangen einer ersten Nonce von der Tastatur an einer Komponente;
Senden einer zweiten Nonce (412) von der Komponente (202) zu der Tastatur;
Schaffen eines ersten Anfangswertes und eines zweiten Anfangswertes durch Anwenden von Triple-DES und Cipher Block Chaining auf eine Kombination aus der ersten und der zweiten Nonce unter Verwendung eines Schlüssels sowie eines dritten Anfangswertes, der sowohl der Tastatur als auch der Komponente bekannt ist;
Empfangen einer Vielzahl von Daten, die mit Triple-DES und Cipher Block Chaining unter Verwendung des Schlüssels sowie des ersten Anfangswertes verschlüsselt worden sind, an einer Komponente von der Tastatur, wobei jeder separate von der Tastatur empfangene Tastenanschlag in einem separaten der Vielzahl von Datenelementen enthalten ist und jedes der Vielzahl von Datenelementen unter Verwendung eines separaten Blocks des Triple-DES sowie Cipher Block Chaining verschlüsselt wird;
Entschlüsseln der Vielzahl von Datenelementen auf Basis des ersten Anfangswertes und des Schlüssels.

2. Verfahren nach Anspruch 1, das des Weiteren umfasst:

   Empfangen einer Vielzahl von Nachrichten-Authentifizierungscodes, die der Vielzahl von Datenelementen entsprechen, von der Tastatur (162) an der Komponente (202), wobei die Nachrichten-Authentifizierungscodes mit Triple-DES und Cipher Block Chaining unter Verwendung des Schlüssels und des zweiten Anfangswertes geschaffen worden sind, der sich von dem ersten Anfangswert unterscheidet, und der zweite Anfangswert sowohl der Komponente als auch der Tastatur bekannt ist;
   Verifizieren der Vielzahl von Datenelementen unter Verwendung der Vielzahl von Nachrichten-Authentifizierungscodes.

3. Verfahren nach Anspruch 2, wobei Schaffen eines ersten Anfangswertes und eines zweiten Anfangswertes des Weiteren umfasst:

   Schaffen des ersten Anfangswertes durch Anwenden von Triple-DES sowie Cipher Block Chaining auf eine Kombination aus der ersten Nonce und der zweiten Nonce unter Verwendung des Schlüssels sowie eines dritten Anfangwertes, der sowohl der Tastatur als auch der Komponente bekannt ist; und
   Schaffen des zweiten Anfangswertes durch Anwenden von Triple-DES sowie Cipher Block Chaining auf eine Kombination aus der ersten Nonce und der zweiten Nonce unter Verwendung des Schlüssels sowie eines vierten Anfangswertes, der sowohl der Tastatur als auch der Komponente bekannt ist.

4. Verfahren nach Anspruch 1, wobei die Vielzahl von Daten über einen Kanal empfangen werden, dessen Verhaltensintegrität von der Komponente nicht vertraut wird.

5. Verfahren nach Anspruch 4, wobei die Komponente (202) ein erstes Betriebssystem umfasst, das auf einer Computereinrichtung zusammen mit einem zweiten Betriebssystem läuft, das erste Betriebssystem wenigstens in bestimmter Hinsicht dem Verhalten des zweiten Betriebssystems nicht vertraut und die Tastatur über einen Treiber, der von dem zweiten Betriebssystem gesteuert wird, mit dem ersten Betriebssystem kommuniziert.

6. Verfahren nach Anspruch 1, wobei die Tastatur eine USB-Tastatur umfasst.

7. Verfahren nach Anspruch 6, wobei die unter Verwendung von Triple-DES sowie Cipher Block Chaining verschlüsselten Daten in Blöcken eine vorgegebene Größe haben und die Tastatur Daten in Blöcken der vorgegebenen Größe überträgt.

8. Verfahren nach Anspruch 1, wobei jedes der Vielzahl von Datenelementen auf Basis eines einzelnen an der Tastatur empfangenen Tastenanschlags erzeugt wird.

9. Computerlesbares Medium, auf dem durch Computer ausführbare Befehle zum Durchführen eines Verfahrens zum sicheren Empfangen von Eingabe an einer Software-Komponente (202) von einer Tastatur (162) kodiert sind, wobei das Verfahren umfasst:

   Empfangen einer ersten Nonce (412) von der Tastatur an einer Software-Komponente;
   Senden einer zweiten Nonce (414) an die Tastatur von der Software-Komponente;
   Schaffen eines ersten Anfangswertes und eines zweiten Anfangswertes durch Anwenden von Triple-DES sowie Cipher Block Chaining auf eine Kombination aus der ersten und der zweiten Nonce unter Verwendung eines Schlüssels sowie eines dritten Anfangswertes, der sowohl der Tastatur als auch der Komponente bekannt ist;
   Empfangen einer Vielzahl verschlüsselter Tastenanschläge von der Tastatur an der Software-Komponente, wobei die verschlüsselten Tastenanschläge an der Tastatur durch Verschlüsseln an der Tastatur empfangener Eingabe-Tastenanschläge mit Triple-DES und Cipher Block Chaining unter Verwendung des Schlüssels sowie des ersten Anfangswertes geschaffen worden sind, jeder separate von der Tastatur empfangene Tastenanschlag in einem separaten der Vielzahl von Datenelementen enthalten ist und jedes der Vielzahl von Datenelementen unter Verwendung eines separaten Blocks des Triple-DES und Cipher Block Chaining verschlüsselt wird; und
   Entschlüsseln der Vielzahl verschlüsselter Tastenanschläge unter Verwendung des Schlüssels und des ersten Anfangswertes an der Software-Komponente.

10. Computerlesbares Medium nach Anspruch 9, wobei die Software-Komponente (202) ein erstes Betriebssystem umfasst, das auf einer Computereinrichtung zusammen mit einem zweiten Betriebssystem läuft, das erste Betriebs-

system wenigstens in bestimmter Hinsicht dem Verhalten des zweiten Betriebssystems nicht vertraut und die Tastatur über einen Treiber, der von dem zweiten Betriebssystem gesteuert wird, mit dem ersten Betriebssystem kommuniziert.

11. Computerlesbares Medium nach Anspruch 9, wobei das Verfahren des Weiteren umfasst:

Empfangen einer Vielzahl von Nachrichten-Authentifizierungscodes, die der Vielzahl verschlüsselter Tastenanschläge entsprechen, von der Tastatur (162) an der Software-Komponente (202), wobei die Nachrichten-Authentifizierungscodes mit Triple-DES sowie Cipher Block Chaining unter Verwendung des Schlüssels sowie des zweiten Anfangswertes geschaffen worden sind, der sich von dem ersten Anfangswert unterscheidet, und der zweite Anfangswert sowohl der Komponente als auch der Tastatur bekannt ist;
Verifizieren der Vielzahl verschlüsselter Tastenanschläge unter Verwendung der Vielzahl von Nachrichten-Authentifizierungscodes..

12. Computerlesbares Medium nach Anspruch 11, das des Weiteren umfasst:

Schaffen des ersten Anfangswertes durch Anwenden von Triple-DES sowie Cipher Block Chaining auf eine Kombination aus der ersten Nonce (412) und der zweiten Nonce (414) unter Verwendung des Schlüssels sowie eines dritten Anfangwertes, der sowohl der Tastatur als auch der Software-Komponente (202) bekannt ist; und
Schaffen des zweiten Anfangswertes durch Anwenden von Triple-DES sowie Cipher Block Chaining auf eine Kombination aus der ersten Nonce und der zweiten Nonce unter Verwendung des Schlüssels und eines vierten Anfangswertes, der sowohl der Tastatur als auch der Komponente bekannt ist.

13. Computerlesbares Medium nach Anspruch 9, wobei die unter Verwendung von Triple-DES sowie Cipher Block Chaining verschlüsselten Daten in Blöcken eine vorgegebene Größe haben und die Tastatur Daten in Blöcken der vorgegebenen Größe überträgt.

14. Tastatur (162), die umfasst:

einen oder mehrere Speicherorte, die einen ersten Anfangswert und einen Schlüssel speichern;
eine Verschlüsselungskomponente (202), die so eingerichtet ist, dass sie eine erste Nonce (412) von dem Empfänger über die Kommunikations-Schnittstelle empfängt, eine zweite Nonce (414) zu dem Empfänger über die Kommunikations-Schnittstelle sendet und den ersten Anfangswert durch Anwenden von Triple-DES sowie Cipher Block Chaining auf eine Kombination aus der ersten Nonce und der zweiten Nonce unter Verwendung des Schlüssels sowie eines zweiten Anfangswertes schafft, der sowohl der Tastatur als auch der Komponente bekannt ist, wobei die Verschlüsselungskomponente an der Tastatur empfangene Eingabedaten mit Triple-DES sowie Cipher Block Chaining unter Verwendung des Schlüssels sowie des ersten Anfangswertes verschlüsselt, verschlüsselte Daten auf Basis der Eingabedaten geschaffen werden, jedes einzelne der Eingabe-Datenelemente repräsentativ für einen separaten an der Tastatur empfangenen Tastenanschlag ist, jedes einzelne der Eingabe-Datenelemente unter Verwendung eines separaten Blocks des Triple-DES sowie Cipher Block Chaining verschlüsselt wird; und
eine Kommunikations-Schnittstelle, die die verschlüsselten Daten zu einer außerhalb der Tastatur befindlichen Einrichtung überträgt, wobei die verschlüsselten Daten für einen Empfänger bestimmt sind, der den ersten Anfangswert und den Schlüssel kennt.

15. Tastatur (162) nach Anspruch 14, wobei die Verschlüsselungskomponente des Weiteren eine Vielzahl von Nachrichten-Authentifizierungscodes schafft, die den verschlüsselten Daten oder den Eingangsdaten entsprechen, und die Nachrichten-Authentifizierungscodes mit Triple-DES sowie Cipher Block Chaining unter Verwendung des Schlüssels sowie eines Anfangswertes geschaffen worden sind, der sich von dem ersten Anfangswert unterscheidet.

16. Tastatur (162) nach Anspruch 14, wobei die Tastatur eine USB-Tastatur umfasst.

17. Tastatur (162) nach Anspruch 14, wobei die Kommunikations-Schnittstelle die verschlüsselten Daten zu einem Treiber überträgt, der von einem ersten Betriebssystem gesteuert wird, das auf der Einrichtung läuft, und daraufhin die verschlüsselten Daten zu dem Empfänger übertragen werden, der Empfänger ein zweites Betriebssystem, das auf der Einrichtung läuft, oder ein Programm ist, das unter dem zweiten Betriebssystem läuft, und das zweite Betriebssystem wenigstens in bestimmter Hinsicht dem Verhalten des ersten Betriebssystems nicht vertraut.

Revendications

1. Procédé de communication avec un clavier (162), comportant les étapes de:

   réception du clavier, sur un composant, d'une première valeur à usage unique ;
   émission vers le clavier, à partir du composant (202), d'une seconde valeur à usage unique (412) ;
   création d'une première valeur initiale et d'une seconde valeur initiale par application d'un chaînage triple DES et chiffrement par bloc à une combinaison de ladite première valeur à usage unique et de ladite seconde valeur à usage unique, en utilisant une clé et une troisième valeur initiale connue à la fois du clavier et du composant ;
   réception, sur un composant, de plusieurs données cryptées par application d'un chaînage triple DES et chiffrement par bloc en utilisant la clé et la première valeur initiale provenant du clavier, chaque frappe distincte sur une touche, reçue dudit clavier, étant incluse dans une donnée distincte desdites données, chacune desdites données étant cryptée en utilisant un bloc distinct dudit chaînage triple DES et chiffrement par bloc ;
   décryptage desdites données en se basant sur ladite première valeur initiale et ladite clé.

2. Procédé selon la revendication 1, comportant en outre les étapes de :

   réception, sur le composant (202), de plusieurs codes d'authentification de message correspondant aux dites données provenant du clavier (162), lesdits codes d'authentification de message ayant été créés par application d'un chaînage triple DESS et chiffrement par bloc en utilisant ladite clé et la seconde valeur initiale, différente de ladite première valeur initiale, ladite seconde valeur initiale étant connue à la fois du composant et du clavier ;
   vérification desdites données en utilisant lesdits codes d'authentification de message.

3. Procédé selon la revendication 2, dans lequel l'étape de création d'une première valeur initiale et d'une seconde valeur initiale comporte en outre les étapes de :

   création de la première valeur initiale par application d'un chaînage triple DES et chiffrement par bloc à une combinaison de la première valeur à usage unique et de la seconde valeur à usage unique, en utilisant la clé et une troisième valeur initiale connue à la fois du clavier et du composant ; et
   création de la seconde valeur initiale par application d'un chaînage triple DES et chiffrement par bloc à une combinaison de la première valeur à usage unique et de la seconde valeur à usage unique, en utilisant la clé et une quatrième valeur initiale connue à la fois du clavier et du composant.

4. Procédé selon la revendication 1, dans lequel lesdites données sont reçues au moyen d'un canal dont l'intégrité de comportement n'est pas considérée comme digne de confiance par le composant.

5. Procédé selon la revendication 4, dans lequel le composant (202) comporte un premier système d'exploitation, qui s'exécute sur un dispositif informatique parallèlement à un second système d'exploitation, le premier système d'exploitation se défiant, au moins à certains égards, du comportement du second système d'exploitation, le clavier communiquant avec le premier système d'exploitation au moyen d'un pilote contrôlé par le second système d'exploitation.

6. Procédé selon la revendication 1, dans lequel le clavier est constitué d'un clavier USB.

7. Procédé selon la revendication 6, dans lequel ledit chaînage triple DES et chiffrement de bloc crypte les données par blocs possédant une taille prédéterminée, et dans lequel le clavier communique des données par blocs de ladite taille prédéterminée.

8. Procédé selon la revendication 1, dans lequel chacune des données est générée en se basant sur une frappe de touche individuelle reçue sur le clavier.

9. Support lisible par ordinateur codé au moyen d'instructions exécutables par ordinateur pour mettre en oeuvre un procédé consistant à recevoir de manière sécurisée une entrée sur un composant logiciel (202) à partir d'un clavier 162, le procédé comportant les étapes de :

   réception, sur un composant logiciel, d'une première valeur à usage unique (412) en provenance du clavier ;
   émission, à partir du composant logiciel, d'une seconde valeur à usage unique (414) vers le clavier ;
   création d'une première valeur initiale et une seconde valeur initiale par application d'un chaînage triple DES

et chiffrement de bloc à une combinaison de ladite première valeur à usage unique et de ladite seconde valeur à usage unique, en utilisant une clé et une troisième valeur initiale connue à la fois du clavier et du composant ;
réception, sur le composant logiciel, de plusieurs frappes de touche cryptées provenant du clavier, les frappes de touche cryptées ayant été créées sur le clavier en cryptant des frappes de touche d'entrée reçues sur le clavier, par un chaînage triple DES et chiffrement de bloc en utilisant la clé et la première valeur initiale, chaque frappe de touche distincte reçue dudit clavier étant incluse dans une donnée séparée desdites données, chacune desdites données étant cryptée en utilisant un bloc séparé dudit chaînage triple DES et chiffrement de bloc ; et
sur le composant logiciel, décryptage desdites frappes de touche cryptées en utilisant la clé et la première valeur initiale.

10. Support lisible par ordinateur selon la revendication 9, dans lequel le composant logiciel comporte (202) un premier système d'exploitation, qui s'exécute sur un dispositif informatique parallèlement à un second système d'exploitation, le premier système d'exploitation se défiant, au moins à certains égards, du comportement du second système d'exploitation, le clavier communiquant avec le premier système d'exploitation au moyen d'un pilote contrôlé par le second système d'exploitation.

11. Support lisible par ordinateur selon la revendication 9, dans lequel le procédé comporte en outre les étapes de :

réception, sur le composant logiciel (202), de plusieurs codes d'authentification de message, correspondant auxdites frappes de touche cryptées issues du clavier (162), lesdits codes d'authentification de message ayant été créés par un chaînage triple DES et chiffrement de bloc en utilisant ladite clé et la seconde valeur initiale différente de ladite première valeur initiale, ladite seconde valeur initiale étant connue à la fois du composant logiciel et du clavier ;
vérification desdites frappes de touche cryptées en utilisant lesdits codes d'authentification de message.

12. Support lisible par ordinateur selon la revendication 11, dans lequel l'étape de création d'une première valeur initiale et une seconde valeur initiale comporte en outre les étapes de :

création de la première valeur initiale par application d'un chaînage triple DES et chiffrement de bloc à une combinaison de la première valeur à usage unique (412) et de la seconde valeur à usage unique (414), en utilisant la clé et la troisième valeur initiale connue à la fois du clavier et du composant logiciel (202) ; et de
création la seconde valeur initiale par application d'un chaînage triple DES et chiffrement de bloc à une combinaison de la première valeur initiale et de la seconde valeur initiale, en utilisant la clé et une quatrième valeur initiale connue à la fois du clavier et du composant logiciel.

13. Support lisible par ordinateur selon la revendication 9, dans lequel ledit chaînage triple DES et chiffrement de bloc crypte des données par blocs possédant une taille prédéterminée, et dans lequel le clavier communique des données par blocs de ladite taille prédéterminée.

14. Clavier (162) comportant :

un ou deux emplacements de stockage destinés à stocker une première valeur initiale et une clé ;
un composant de cryptage (202), conçu pour recevoir une première valeur à usage unique (412) du destinataire au moyen de l'interface de communication, émettre une seconde valeur à usage unique (414) vers le destinataire au moyen de l'interface de communication, et créer la première valeur initiale par application d'un chaînage triple DES et chiffrement de bloc à une combinaison de la première valeur à usage unique et de la seconde valeur à usage unique, en utilisant la touche et une seconde valeur initiale connue à la fois du clavier et du composant, dans lequel ledit composant de cryptage crypte des données d'entrée reçues sur le clavier par un chaînage triple DES et chiffrement de bloc en utilisant ladite clé et ladite première valeur initiale, ce qui provoque la création de données cryptées, en se basant sur lesdites données d'entrée, chacune desdites données d'entrée étant représentative d'une frappe de touche distincte reçue sur ledit clavier, chacune desdites données d'entrée étant cryptée en utilisant un bloc distinct dudit chaînage tripe DES et chiffrement de bloc ; et
une interface de communication qui communique lesdites données cryptées à un dispositif extérieur au clavier, lesdites données cryptées étant destinées à un destinataire qui connaît ladite première valeur initiale et ladite clé.

15. Clavier (162) selon la revendication 14, dans lequel le composant de cryptage crée en outre plusieurs codes d'authentification de message correspondant aux données cryptées ou aux données d'entrée, lesdits codes d'authentification de message ayant été créés par un chaînage triple DES et chiffrement de bloc en utilisant ladite clé et une valeur initiale différente de ladite première valeur initiale.

16. Clavier (162) selon la revendication 14, dans lequel le clavier est constitué d'un clavier USB.

17. Clavier (162) selon la revendication 14, dans lequel l'interface de communication communique lesdites données cryptées à un pilote contrôlé par un premier système d'exploitation qui s'exécute sur ledit dispositif, après quoi lesdites données cryptées sont communiquées audit destinataire, ledit destinataire étant un second système d'exploitation qui s'exécute sur ledit dispositif ou un programme qui s'exécute sous ledit second système d'exploitation, ledit second système d'exploitation se défiant, au moins à certains égards, du comportement du premier système d'exploitation.

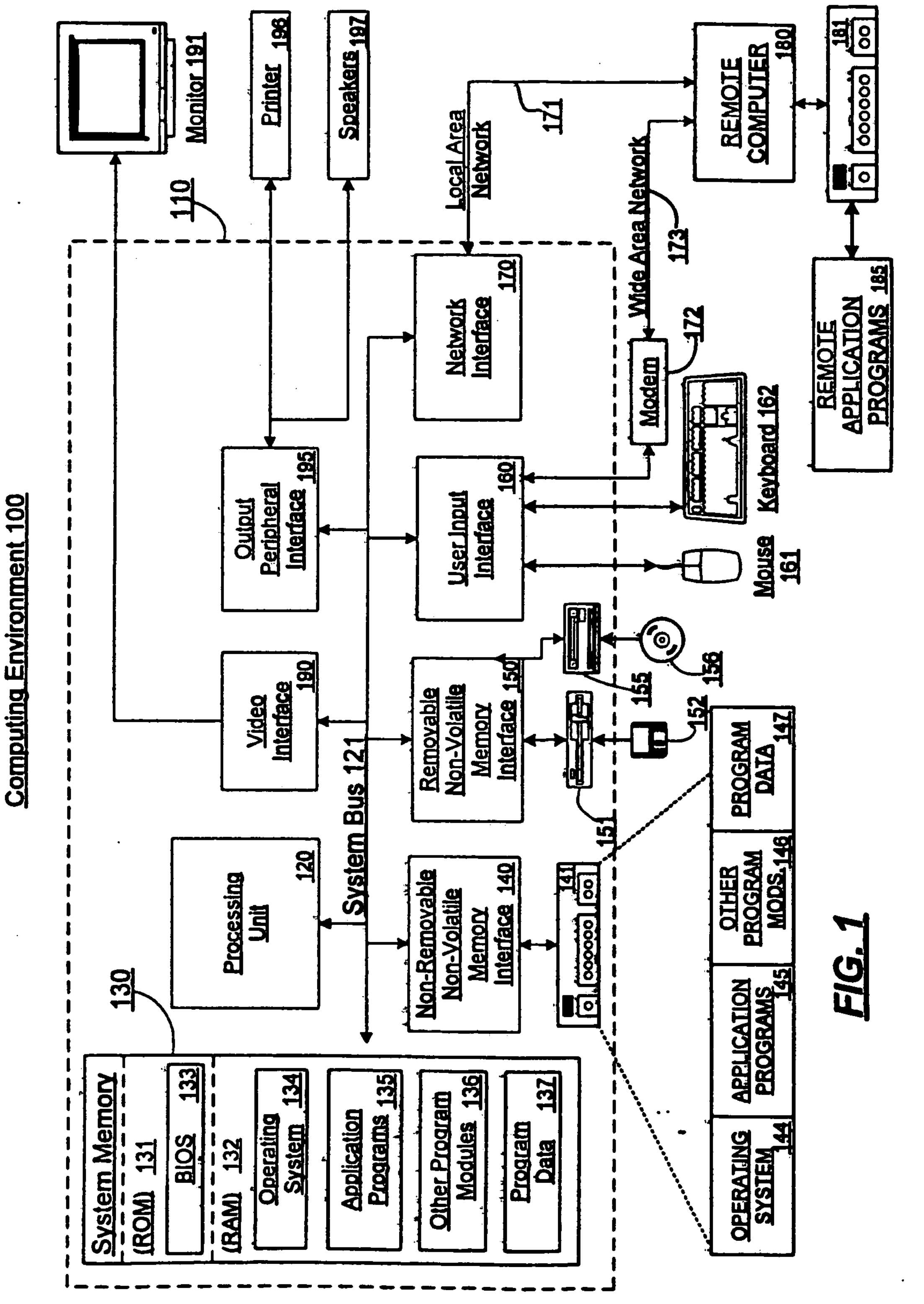
Computing Environment 100
110
130
System Memory
(ROM) 131
BIOS 133
(RAM) 132
Operating System 134
Application Programs 135
Other Program Modules 136
Program Data 137
Processing Unit 120
System Bus 121
Non-Removable Non-Volatile Memory Interface 140
141
Removable Non-Volatile Memory Interface 150
151
152
155
156
Video Interface 190
Output Peripheral Interface 195
User Input Interface 160
Network Interface 170
Monitor 191
Printer 196
Speakers 197
Local Area Network
171
Wide Area Network
173
Modem
172
Keyboard 162
Mouse 161
REMOTE COMPUTER 180
181
REMOTE APPLICATION PROGRAMS 185
OPERATING SYSTEM 144
APPLICATION PROGRAMS 145
OTHER PROGRAM MODS. 146
PROGRAM DATA 147

FIG. 1

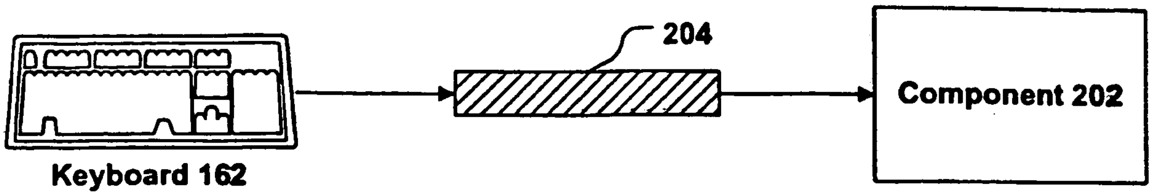

FIG. 2

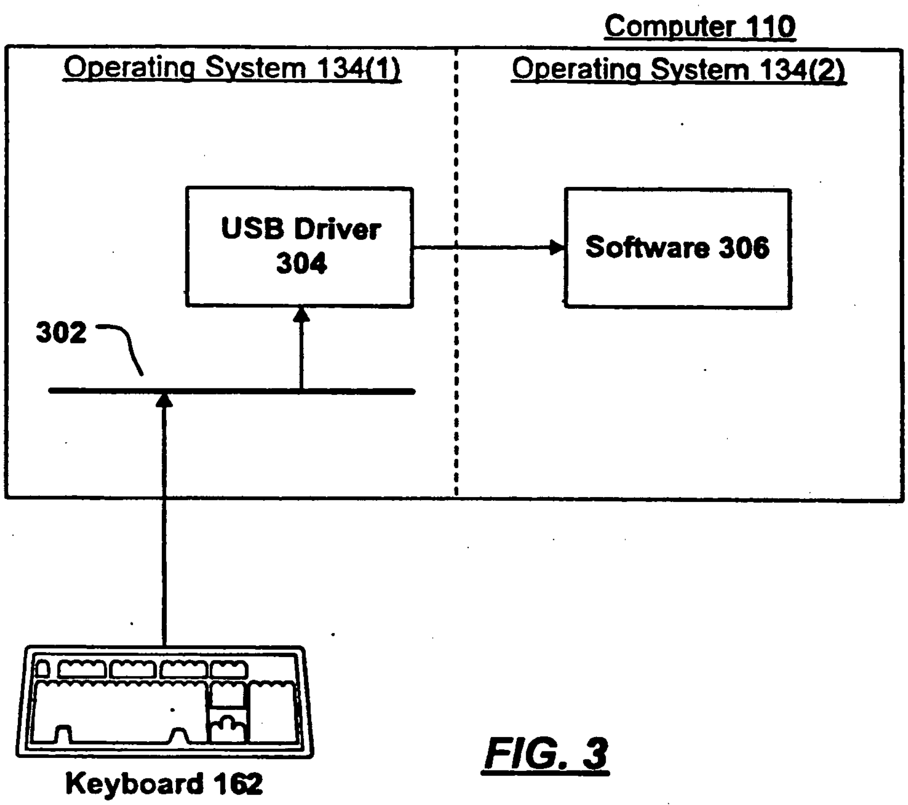

FIG. 3

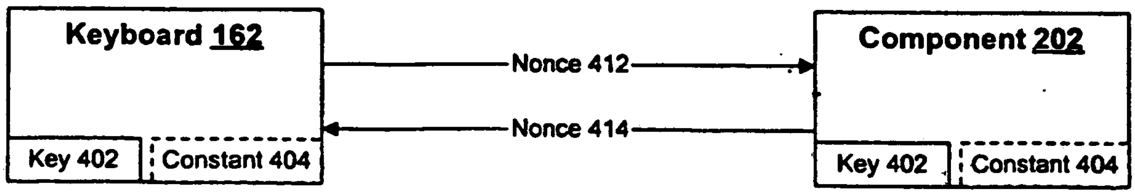

FIG. 4

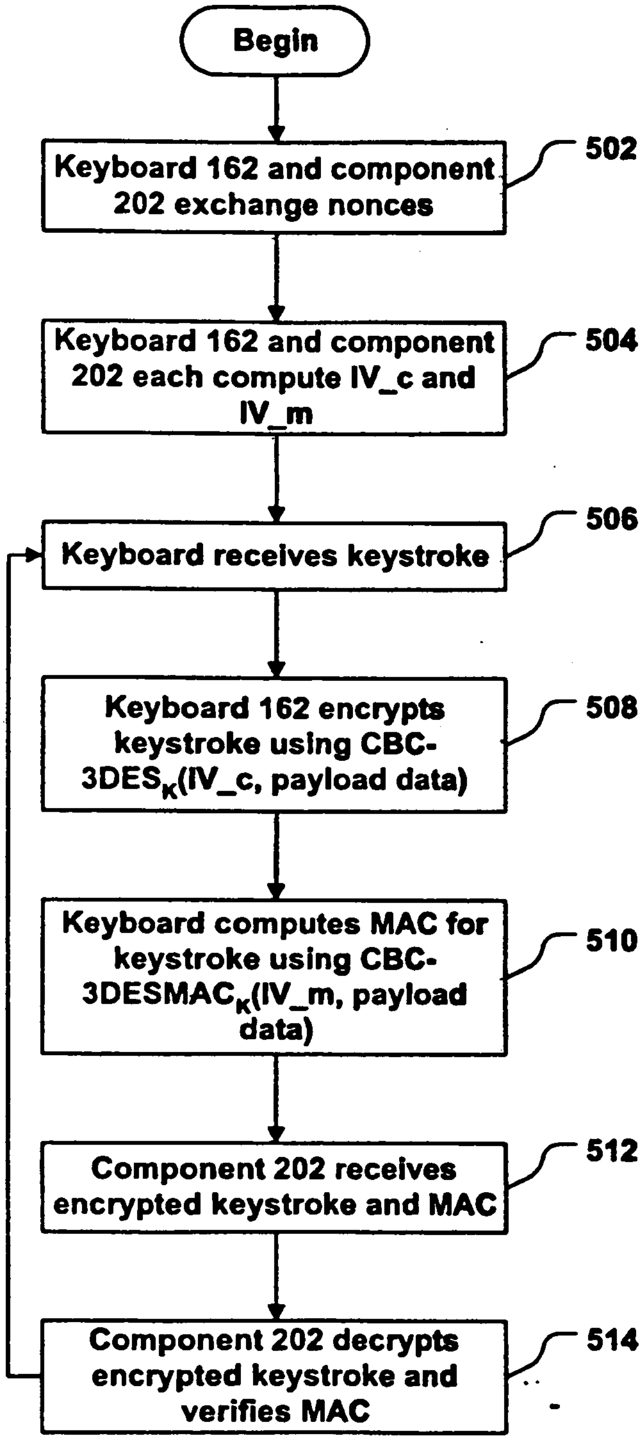
Begin
Keyboard 162 and component 202 exchange nonces
502
Keyboard 162 and component 202 each compute IV_c and IV_m
504
Keyboard receives keystroke
506
Keyboard 162 encrypts keystroke using CBC-3DES$_K$(IV_c, payload data)
508
Keyboard computes MAC for keystroke using CBC-3DESMAC$_K$(IV_m, payload data)
510
Component 202 receives encrypted keystroke and MAC
512
Component 202 decrypts encrypted keystroke and verifies MAC
514

FIG. 5

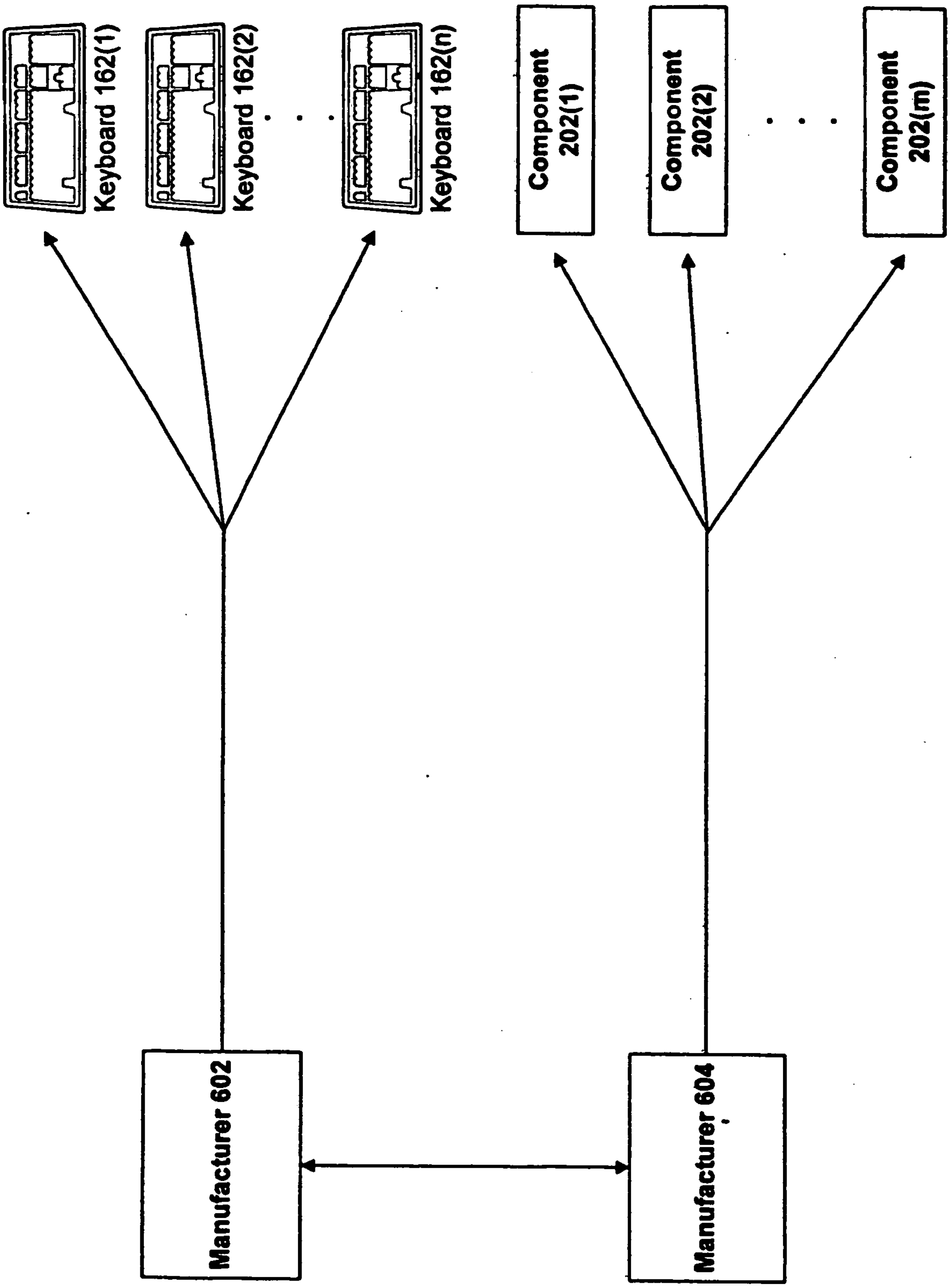
Keyboard 162(1)
Keyboard 162(2)
Keyboard 162(n)
Component 202(1)
Component 202(2)
Component 202(m)
Manufacturer 602
Manufacturer 604

FIG. 6

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- EP 1286242 A **[0006]**

Non-patent literature cited in the description

- **G. Treat.** Keyboard encryption. *IEEE Potentials,* August 2002, vol. 21 (3), 40-42 **[0007]**