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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 January 2005 (20.01.2005)

PCT

(10) International Publication Number
WO 2005/006642 A1

(51) International Patent Classification⁷: H04L 9/00

(21) International Application Number: PCT/US2003/033380

(22) International Filing Date: 20 October 2003 (20.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/464,891 19 June 2003 (19.06.2003) US

(71) Applicant: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(72) Inventors: PEYRAVIAN, Mohammad; 304 Oxcroft Street, Cary, NC 27519 (US). CRAFT, David, J.; 114 Bluff Park Circle, Austin, TX 78746-4300 (US). HOFSTEE, Harm, P.; 704 Pressler Street, Austin, TX 78703 (US). BROKENSHIRE, Daniel, A.; 3567 Ashmere Loop, Round Rock, TX 78681 (US).

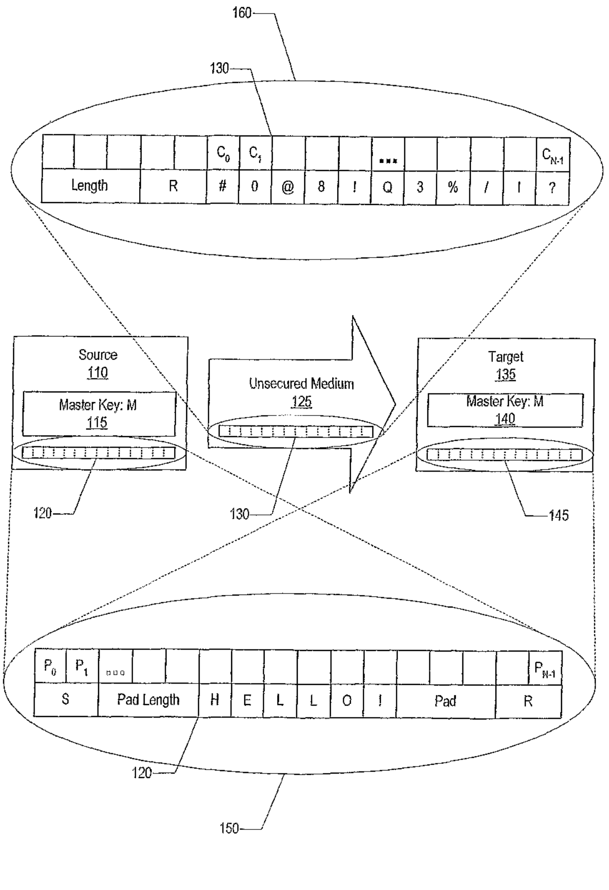
(74) Agent: SCHNURMANN, H., Daniel; International Business Machines Corporation, Dept. 18G BLDG. 300/482, 2070 Route 52, Hopewell Jonction, NY 12533 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR ENCRYPTING AND VERIFYING MESSAGES USING THREE-PHASE ENCRYPTION



(57) Abstract: A method and system for encrypting and verifying the integrity of a message using a three-phase encryption process is provided. A source having a secret master key that is shared with a target receives the message and generates a random number. The source then generates: a first set of intermediate values from the message and the random number; a second set of intermediate values from the first set of values; and a cipher text from the second set of values. At the three phases, the values are generated using the encryption function of a block cipher encryption/decryption algorithm. The random number and the cipher text are transmitted to the target, which decrypts the cipher text by reversing the encryption process. The target verifies the integrity of the message by comparing the received random number with the random number extracted from the decrypted cipher text.

European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

System and Method for Encrypting and Verifying Messages Using Three-Phase Encryption

Background of the Invention

The present invention relates in general to a system and method for encrypting, decrypting, and verifying the integrity of messages. In particular, the present invention relates to a system and method for using a three-phase encryption process to encrypt, decrypt, and verify the integrity of a message.

The Internet provides an efficient and inexpensive means of communication between large numbers of users. The same infrastructure can be shared among the different users; it is unnecessary for each pair of communicating users to establish a separate channel of communication as is the case, for example, with users of standard telephones and fax machines. The sharing of the channels of communication provided by the Internet significantly increases the probability of intercepted communication, eavesdropping, tampering with the messages, etc. Thus, it has become increasingly important to have available means of communication that provide an efficient way of securing the transmission of messages between a source and a target over shared channels of communication such as the Internet. The most common method of secured communication is to encrypt the message at the source in such a way that practically only the target can decrypt the message.

Many types of encryption/decryption have been developed to address the need for secured communications. There are two main types of encryption techniques: secret key encryption and private/public key encryption. In secret key encryption, the same secret key is used both for the encryption of the message at the source and the decryption of the message at the target. An example of secret key encryption is the Data Encryption Standard (DES). In public/private key encryption, each user has a private key (which is kept secret from the other users) and a public key (which each user publicly distributes). The two keys are mathematically related in such a way that a source uses the target's public key to encrypt a message such that practically only the target can decrypt the message.

A good encryption standard must be computationally efficient for the source and the target, and at the same time, the standard must be not be easy to "break" using cryptanalysis: the art and science of breaking encryption algorithm. It is also important for the encryption standard to provide means for verifying the integrity of a message—whether a message was altered during its transmission through an unsecured medium.

What is needed, therefore, is a system and method that could provide an efficient encryption/decryption standard between a source and a target while not being susceptible to cryptanalysis. The system and method should also provide the capability to verify the integrity of a transmitted message to a high degree of probability.

Summary

It has been discovered that the aforementioned challenges can be addressed by a method and a system for encrypting, decrypting, and verifying the integrity of a message using a three-phase encryption process. The structure of the three-phase encryption process makes cryptanalysis of the algorithm extremely difficult, which significantly reduces the probability of "breaking" the encryption. In addition, the current method and system provide a means for verifying the integrity of a transmitted message by comparing, at the target, a received control number to a decrypted control number.

The source and target share a secret master key, which the source uses to encrypt a message, and after the message is transmitted to the target, the target uses to decrypt the message. The message to be encrypted and transmitted to the target is received by the source in plain-text form. The source first generates a first and second random number and constructs a plain-text envelope comprising: the plain-text message; the first random number; the second random number; a pad field; and a number indicating the length of the pad field. The pad field is generated such that a length of: the plain-text message; the first random number; the second random number; the pad field; and the number indicating the length of the pad field is an integer multiple of a block size of a block cipher algorithm used in the encryption/decryption process. A first, second, and third key are subsequently generated. The three keys are used one at each of the three phases of the encryption process. The keys are generated from the secret master key and the first random number using a decryption function of a block cipher algorithm used in the encryption/decryption process.

At the fist phase of the encryption process, a first set of N intermediate blocks is generated from the N plain-text blocks and the first key using the function of a block cipher algorithm encryption/decryption. At the second phase of the encryption process, a second set of N intermediate blocks is generated from the first set of N intermediate blocks and the second key using the encryption function of the block cipher encryption/decryption algorithm. At the third and final phase of the encryption process, N cipher text blocks are generated from the second set of N intermediate blocks and the third key using the encryption function of the block cipher encryption/decryption algorithm.

A protected-text envelope is then constructed containing: a number indicating the length of the protected-text envelope; the first random value; and the N cipher-text blocks. The protected-text envelope is transmitted to a target over an unsecured medium such as the Internet.

The protected-text envelope is received by the target, which shares the source's secret master key. The target extracts from the protected-text envelope: the number indicating the length of the protected-text envelope; the first random value; and the N cipher-text blocks. The first, second, and third keys are then generated from the extracted first random number and the secret master key using the decryption function of the block cipher encryption/decryption algorithm.

At the fist phase of the decryption process, the second set of the N intermediate blocks is generated from the N cipher-text blocks and the third key using the decryption function of the block cipher encryption/decryption algorithm. At the second phase of the decryption process, second set of the N intermediate blocks is generated from the first set of N intermediate blocks and the second key using the decryption function of the block cipher encryption/decryption algorithm. At the third and final phase of the decryption process, the N plain-text blocks are generated from the first set of N intermediate blocks and the first key using the decryption function of the block cipher encryption/decryption algorithm. The plain-text message is then extracted from the N plain-text blocks of the plain-text envelope.

In order to verify the integrity of the message, a first random number included in the N plain-text blocks is then extracted and compared to the first random number extracted from the protected-text envelope. If the two numbers are not equal the received message is not trusted since it can be concluded that the message was most likely altered during its transmission from the source to the target. If the two numbers are equal, the message can be trusted.

The foregoing is a summary and thus contains, by necessity, simplifications, generalizations, and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of the present invention, as defined solely by the claims, will become apparent in the non-limiting detailed description set forth below.

Brief Description of the Drawings

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

Figure 1 is a block diagram illustrating the encryption of a message at a source, transmission in encrypted form over an unsecured medium, and decryption of the message at a target;

Figure 2 is a block diagram illustrating a system for encrypting a message using a three-phase encryption process;

Figure 3 is a block diagram illustrating a system for decrypting a message using a three-phase decryption process;

Figure 4 is a flowchart illustrating a method for receiving, encrypting, and transmitting a message;

Figure 5 is a flowchart illustrating a method for encrypting a message using a three-phase encryption process;

Figure 6 is a flowchart illustrating a method for receiving encrypted text, decrypting the encrypted text, and extracting a message from the decrypted text;

Figure 7 is a flowchart illustrating a method for decrypting an encrypted message using a three-phase decryption process;

Figure 8 is a flowchart illustrating a method for verifying the integrity of a received and decrypted message; and

Figure 9 illustrates an information handling system that is a simplified example of a computer system capable of performing the operations described herein.

Detailed Description

The following is intended to provide a detailed description of an example of the invention and should not be taken to be limiting of the invention itself. Rather, any number of variations may fall within the scope of the invention defined in the claims following the description.

Figure 1 is a block diagram illustrating encryption of a message at a source, transmission in encrypted form over an unsecured medium, and decryption of the message at a target. Source 110 is configured to receive a plain-text message and construct plain-text envelope 120. In one embodiment, plain-text envelope 120 includes the plain-text message and additional numbers and fields generated at the source. Oval 160 shows a larger image of plain-text envelope 120. In one embodiment, envelope 120 contains N blocks P_0-P_{N-1} that include the random number S, a number indicating the length of the pad field, the message "Hello!", a pad field, and the random number R. Source 110 encrypts plain-text envelope 120 to create N cipher text blocks C_0-C_{N-1} using master key M 115 and the three-phase encryption process. Protected-text envelope 130 is then constructed using the cipher text and other control numbers and transmitted through unsecured medium 125. Oval 160 shows a larger image of protected-text envelope 130. In one embodiment, envelope 130 contains a number indicating the length of the protected-text envelope, the control random number R, and the cipher text.

Target 135 is configured to receive the message and decrypt the extracted cipher text using master key M 140, which is shared between the source and the target. After decryption, the target recovers plain-text envelope 145, which should be identical to plain-text envelope 120 unless the cipher text was altered during transmission through unsecured medium 125.

Figure 2 is a block diagram illustrating a system for encrypting a message using the three-phase encryption process. Encryption device 210 is configured to receive the N blocks $P_0 - P_{N-1}$ of the plain-text envelope P and generate the N blocks $C_0 - C_{N-1}$ of cipher text C. Encryption device 210 includes: N-1 exclusive or modules 220, …, 222, and 224; N-1 exclusive or modules 242, 244, …, 246; N-1 exclusive or modules 256, …, 258, and 260; N encryption modules 212, 214, …, 216, and 218; N encryption modules

226, 228, …, 230, and 232; and N encryption modules 248, 250, …, 252, and 254. The exclusive or modules are configured to generate an output by performing a bitwise "xor" operation on the two inputs to the modules. The encryption modules are configured to generate an output according to the formula

$$Out = \mathrm{Enc}_{K_i}(In).$$

The function Enc() may be the encryption function of any block cipher encryption/decryption algorithm and K_i is one of three keys K_1, K_2, and K_3 generated at the source. The first key is used in encryption modules 212-218, the second key is used in encryption modules 226-232, and the third key is used in encryption modules 248-254. The three keys are generated according to the formula:

$$K_i = \mathrm{Dec}_M(R \oplus i), \quad i = 1,2,3.$$

Dec() may be the decryption function of any block cipher encryption/decryption algorithm, M is the secret master key, and $\oplus$ is the "xor" operator.

Initially, block P_0 is input into encryption module 212 to generate intermediate block A_0. Block A_0 and block P_1 are then input into exclusive or module 220 and the output from exclusive or module 220 is input into encryption module 214 to generate intermediate block A_1. The process repeats until block A_{N-2} and block P_{N-1} are input into exclusive or module 224 and the output from exclusive or module 224 is input into encryption module 218 to generate intermediate block A_{N-1} to complete the first phase of the encryption process.

The second phase of the encryption process begins with intermediate block A_{N-1} being input into encryption module 232 to generate intermediate block B_{N-1}. Block A_{N-2} is then input into encryption module 230 and the output from encryption module 230 and block A_{N-1} are input into exclusive or module 246 to generate intermediate block B_{N-2}. This process repeats until block A_1 is input into encryption module 226 and the output from encryption module 226 and block A_0 are input into exclusive or module 242 to generate intermediate block B_0 to complete the second phase of the encryption process.

The third phase of the encryption process begins with intermediate block B_0 being input into encryption module 248 to generate cipher text block C_0. Block B_1 is then input into encryption module 250 and the output from encryption module 250 and block B_0 are input into exclusive or module 256 to generate cipher text block C_1. This process repeats until block B_{N-1} is input into encryption module 254 and the output from encryption module 254 and block B_{N-2} are input into exclusive or module 260 to generate cipher text block C_{N-1} to complete the third phase of the encryption process.

Figure 3 is a block diagram illustrating a system for decrypting a message using a three-phase decryption process. Decryption device 310 is configured to receive the N blocks $C_0 - C_{N-1}$ of cipher text C and generate the N blocks $P_0 - P_{N-1}$ of the plain-text envelope P. Decryption device 310 includes: N-1 exclusive or modules 320, ..., 322, and 324; N-1 exclusive or modules 334, 236, ..., and 338; N-1 exclusive or modules 348, ..., 350, and 352; N decryption modules 312, 314, ..., 316, and 318; N decryption modules 326, 328, ..., 330, and 332; and N decryption modules 340, 342, ..., 344, and 346. The exclusive or modules are configured to generate an output by performing a bitwise "xor" operation on the two inputs to the modules. The decryption modules are configured to generate an output according to the formula:

$$Out = \mathrm{Dec}_{K_i}(In).$$

The function Enc() may be the decryption function of any block cipher algorithm and K_i is one of three keys K_1, K_2, and K_3 generated by the target. The third key is used in decryption modules 312-318, the second key is used in decryption modules 326-332, and the first key is used in decryption modules 340-346. The three keys are generated according to the formula:

$$K_i = \mathrm{Dec}_M(R \oplus i), \quad i = 1,2,3.$$

Dec() may be the decryption function of any block cipher encryption/decryption algorithm, M is the secret master key, and $\oplus$ is the "xor" operator.

Initially, block C_0 is input into decryption module 312 to generate intermediate block B_0. Block B_0 and block C_1 are then input into exclusive or module 320 and the output from exclusive or module 320 is input into decryption module 314 to generate intermediate block B_1. The process repeats until block C_{N-2} and block C_{N-1} are input into exclusive or module 324 and the output from exclusive or module 324 is input into decryption module 318 to generate intermediate block B_{N-1} to complete the first phase of the decryption process.

The second phase of the decryption process begins with intermediate block B_{N-1} being input into decryption module 332 to generate intermediate block A_{N-1}. Block B_{N-2} and block A_{N-1} are input into exclusive or module 338 and the output from module 338 is input into decryption module 330 to generate intermediate block A_{N-2}. This process repeats until block A_1 and intermediate block B_0 are input into exclusive or module 334 and the output from exclusive or module 334 is input into decryption module 326 to generate intermediate block A_0 to complete the second phase of the decryption process.

The third phase of the decryption process begins with intermediate block A_0 being input into decryption module 340 to generate plain text block P_0. Block A_1 is then input into decryption module 342 and the output from decryption module 342 and block A_0 are input into exclusive or module 348 to generate plain text block P_1. This process repeats until block A_{N-1} is input into decryption module 346 and the output from decryption module 346 and block A_{N-2} are input into exclusive or module 352 to generate plain text block P_{N-1} to complete the third phase of the decryption process.

Figure 4 is a flowchart illustrating a method for receiving, encrypting, and transmitting a message. Processing begins at 400 whereupon, at step 410, secret master key M is received by the source. The same secret master key M is shared between the source and the target. Secret master key M may be received from storage unit 415. The secret master key may be changed frequently to ensure the key's confidentiality. At step 420, the message to be encrypted may be received from storage unit 425. At step 430, two fixed-size random numbers (S & R) are generated.

A determination is then made as to whether a pad field is required to construct a plain-text envelope at decision 435. The plain-text envelope is formed using fixed-size random number S, a number indicating the length of the pad field (if any), the message to be encrypted, the pad field, and the fixed-size random number R. In an embodiment where a block cipher encryption algorithm is to be used in the encryption of the plain-text envelope, the size (length) of the envelope must be an integral multiple of the cipher's block size. For example, the block size for the AES block cipher algorithm is 128 bits. If the length of the message, S, R, and the number indicating the length of the pad field is an integral multiple of the cipher's block size, decision 435 branches to "yes" branch 445 whereupon processing continues at step 455.

If the length of the message, S, R, and the number indicating the length of the pad field is not an integral multiple of the cipher's block size, decision 435 branches to "no" branch 440 whereupon, at step 450, a pad field containing any arbitrary pattern is generated. The length of the pad field is chosen so as to provide a plain-text envelope having a length that is an integral multiple of the cipher's block size. At step 455, a plain-text envelope P is generated. In one embodiment, envelope P contains: the fixed-size random number S, a number indicating the length of the pad field, the message to be encrypted, the pad field, and the fixed-size random number R. Envelope P contains N equal-size blocks P_0-P_{N-1}.

At step 460, the plain-text envelope P is encrypted using the three-phase encryption process to construct a protected-text envelope. More details of the encryption process are shown in the flowchart of Figure 5. At step 460, the protected-text envelope is transmitted to the target over an unsecured medium such as the Internet.

Figure 5 is a flowchart illustrating a method for encrypting a message using a three-phase encryption process. Processing begins at 500 whereupon at 510, three keys are generated from the random number R and the secret master key using the decryption function of a block cipher algorithm. In one embodiment, the three keys may be generated according to the formula:

$$K_i = \mathrm{Dec}_M(R \oplus i), \quad i = 1,2,3\,.$$

Dec() may be the decryption function of any block cipher encryption/decryption algorithm, M is the secret master key, and $\oplus$ is the "xor" operator.

At step 515, the first of three phases of the three-phase encryption is performed. N intermediate blocks A (A_0-A_{N-1}) are generated from the N blocks of the plain-text envelope and the first key using an encryption function according to the formulas:

$$A_i = \begin{cases} \mathrm{Enc}_{K_1}(P_i) & i = 0 \\ \mathrm{Enc}_{K_1}(P_i \oplus A_{i-1}) & i = 1,2,\ldots,N-1 \end{cases}.$$

Enc() may an encryption function of a block cipher encryption/decryption algorithm.

At step 520, the second of three phases of the three-phase encryption is performed. N intermediate blocks B (B_0-B_{N-1}) are generated from the N intermediate A blocks and the second key K_2 using an encryption function according to the formulas:

$$B_i = \begin{cases} \mathrm{Enc}_{K_2}(A_i) & i = N-1 \\ A_{i+1} \oplus \mathrm{Enc}_{K_2}(A_i) & i = N-2, N-3, \ldots, 0 \end{cases}.$$

At step 525, the third and final phase of the three-phase encryption is performed. N cipher text blocks C (C_0-C_{N-1}) are generated from the N intermediate blocks B and the third key K_3 using an encryption function according to the formulas:

$$C_i = \begin{cases} \mathrm{Enc}_{K_3}(B_i) & i = 0 \\ B_{i-1} \oplus \mathrm{Enc}_{K_3}(B_i) & i = 1, 2, \ldots, N-1 \end{cases}.$$

A protected-text envelope is then constructed. In one embodiment, the protected-text envelope contains: the length of the envelope, the random number R, and the cipher text blocks C. Processing ends at 599.

Figure 6 is a flowchart illustrating a method for receiving encrypted text, decrypting the encrypted text, and extracting a message from the decrypted text. Processing begins at 600 whereupon, at step 610, a protected-text envelope is received through an unsecured medium such as the Internet. In one embodiment, the protected-text envelope contains: the length of the envelope, a random number R, and N cipher text blocks C. The cipher text blocks typically contain a message in encrypted form. At step 615, the length of the envelope, the random number R, and the cipher text blocks C are extracted from the protected-text envelope. At step 620, the cipher text blocks are decrypted to recover N blocks of a plain-text envelope. More details on the decryption are provided in the flowchart of Figure 7. In one embodiment, the plain-text envelope contains: a fixed-size random number R, a number representing the length of a pad field contained in the envelope, a message, the pad field, and an additional copy of random number R. The plain-text message is then extracted at step 625. At step 630, the integrity of the message is determined. More details on the integrity determination are provided in the flowchart of Figure 8. Processing ends at 699.

Figure 7 is a flowchart illustrating a method for decrypting an encrypted message using a three-phase process. Processing begins at 700 whereupon at 710, three keys are generated from the extracted random number R and the secret master key M using the decryption function of a cipher algorithm. In one embodiment, the three keys may be generated according to the formula:

$$K_i = \mathrm{Dec}_M(R \oplus i), \quad i = 1,2,3.$$

Dec() may be the decryption function of any block cipher encryption/decryption algorithm, M is the secret master key, and ⊕ is the "xor" operator.

At step 715, the first of three phases of the three-phase decryption is performed. The N intermediate blocks B (B_0-B_{N-1}) are regenerated from the N cipher text blocks C extracted from the received protected-text envelope and the first key K_1 using a decryption function according to the formulas:

$$B_i = \begin{cases} \mathrm{Dec}_{K_3}(C_i) & i = 0 \\ \mathrm{Dec}_{K_3}(C_i \oplus B_{i-1}) & i = 1,2,\ldots,N-1 \end{cases}.$$

At step 720, the second of three phases of the three-phase decryption is performed. The N intermediate blocks A (A_0-A_{N-1}) are regenerated from the N intermediate B blocks and the second key K_2 using a decryption function according to the formulas:

$$A_i = \begin{cases} \mathrm{Dec}_{K_2}(B_i) & i = N-1 \\ \mathrm{Dec}_{K_2}(B_i \oplus A_{i+1}) & i = N-2, N-3, \ldots, 0 \end{cases}.$$

At step 725, the third and final phase of the three-phase decryption is performed. The N blocks of the plain-text envelope are (P_0-P_{N-1}) are regenerated from the N intermediate A blocks and the third key K_3 using an encryption function according to the formula:

$$P_i = \begin{cases} \mathrm{Dec}_{K_1}(A_i) & i = 0 \\ A_{i-1} \oplus \mathrm{Dec}_{K_1}(A_i) & i = 1,2,\ldots,N-1 \end{cases}.$$

Figure 8 is a flowchart illustrating a method for verifying the integrity of a decrypted message. At step 810, a second copy of the random number R is extracted from the decrypted plain-text envelope P. A first copy of the random number R is extracted from the protected-text envelope C. A determination is then made as to whether the random number R extracted from the plain-text envelope is equal to the random number R extracted from the protected-text envelope R at decision 815. If the two numbers are equal, decision 815 branches to "yes" branch 820 whereupon, at step 830, it is determined that the decrypted message can be trusted. In other words, it is determined that it is highly unlikely that anyone has tampered with the message while the message was being transmitted though the unsecured medium. If the two numbers are not equal, decision 815 branches to "no" branch 825 whereupon, at step 835, it is determined that the decrypted message cannot be trusted. In other words, it is determined that it is highly likely that someone has tampered with the message while the message was being transmitted though the unsecured medium. Processing ends at 899.

Figure 9 illustrates information handling system 901 which is a simplified example of a computer system capable of performing the computing operations described herein. Computer system 901 includes processor 900 which is coupled to host bus 902. A level two (L2) cache memory 904 is also coupled to host bus 902. Host-to-PCI bridge 906 is coupled to main memory 908, includes cache memory and main memory control functions, and provides bus control to handle transfers among PCI bus 910, processor 900, L2 cache 904, main memory 908, and host bus 902. Main memory 908 is coupled to Host-to-PCI bridge 906 as well as host bus 902. Devices used solely by host processor(s) 900, such as LAN card 930, are coupled to PCI bus 910. Service Processor Interface and ISA Access Pass-through 912 provides an interface between PCI bus 910 and PCI bus 914. In this manner, PCI bus 914 is insulated from PCI bus 910. Devices, such as flash memory 918, are coupled to PCI bus 914. In one implementation, flash memory 918 includes BIOS code that incorporates the necessary processor executable code for a variety of low-level system functions and system boot functions.

PCI bus 914 provides an interface for a variety of devices that are shared by host processor(s) 900 and Service Processor 916 including, for example, flash memory 918. PCI-to-ISA bridge 935 provides bus control to handle transfers between PCI bus 914 and

ISA bus 940, universal serial bus (USB) functionality 945, power management functionality 955, and can include other functional elements not shown, such as a real-time clock (RTC), DMA control, interrupt support, and system management bus support. Nonvolatile RAM 920 is attached to ISA Bus 940. Service Processor 916 includes JTAG and I2C busses 922 for communication with processor(s) 900 during initialization steps. JTAG/I2C busses 922 are also coupled to L2 cache 904, Host-to-PCI bridge 906, and main memory 908 providing a communications path between the processor, the Service Processor, the L2 cache, the Host-to-PCI bridge, and the main memory. Service Processor 916 also has access to system power resources for powering down information handling device 901.

Peripheral devices and input/output (I/O) devices can be attached to various interfaces (e.g., parallel interface 962, serial interface 964, keyboard interface 968, and mouse interface 970 coupled to ISA bus 940. Alternatively, many I/O devices can be accommodated by a super I/O controller (not shown) attached to ISA bus 940.

In order to attach computer system 901 to another computer system to copy files over a network, LAN card 930 is coupled to PCI bus 910. Similarly, to connect computer system 901 to an ISP to connect to the Internet using a telephone line connection, modem 975 is connected to serial port 964 and PCI-to-ISA Bridge 935.

While the computer system described in Figure 9 is capable of executing the processes described herein, this computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the processes described herein.

One of the preferred implementations of the invention is an application, namely, a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, on a hard disk drive, or in removable storage such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may be implemented as a computer program product for use in a computer. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those with skill in the art that if a specific number of an introduced claim element is intended, such intent will be explicitly recited in the claim, and in the absence of such recitation no such limitation is present. For a non-limiting example, as an aid to understanding, the following appended claims contain usage of the introductory phrases "at least one" and "one or more" to introduce claim elements. However, the use of such phrases should not be construed to imply that the introduction of a claim element by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an"; the same holds true for the use in the claims of definite articles.

Industrial Applicability

This invention is used in systems requiring transmitting and receiving secure messages which is accomplished by way of encrypting, decrypting and verifying the integrity of the messages.

What is claimed is:

1. A method for encrypting a message, the method comprising:

receiving a plain text;

generating a first key (K_1), a second key (K_2), and a third key (K_3);

generating a first intermediate value from the plain text and the first key (K_1);

generating a second intermediate value from the first intermediate value and the second key (K_2); and

generating a cipher text message from the second intermediate value and the third key ($K_{3)}$.

2. The method of Claim 1, further comprising receiving a secret master key (M).

3. The method of Claim 1, further comprising generating a first random number and a second random number.

4. The method of Claim 1, further comprising generating a pad field such that a length of: the plain-text message; the first random number; the second random number; the pad field; and a number indicating a length of the pad field is an integer multiple of a block size of a block cipher algorithm.

5. The method of Claim 1, further comprising constructing a plain-text envelope comprising: the plain-text message; the first random number; the second random number; the pad field; and the number indicating the length of the pad field.

6. The method of Claim 1, wherein the generating the first intermediate value comprises generating the first intermediate value from the plain-text envelope and the first key.

7. The method of Claim 1, further comprising constructing a protected-text envelope comprising: a number indicating a length of the protected-text envelope; the first random value; and the cipher-text message.

8. The method of Claim 1, further comprising transmitting the protected-text envelope to a target over an unsecured medium.

9. The method of Claim 1, wherein the generating the first key, the second key, and the third key comprises generating the first key, the second key, and the third key from the secret master key and the first random number.

10. The method of Claim 1, wherein the generating the first key, the second key, and the third key comprises generating the first key, the second key, and the third key from the secret master key and the first random number using a decryption function of a block cipher algorithm.

11. The method of Claim 1, wherein the generating the first intermediate value, the second intermediate value, and the cipher-text message comprises using an encryption function of a block cipher algorithm.

12. A method for decrypting a cipher text message, the method comprising:

receiving a cipher text;

generating a first key, a second key, and a third key;

generating a first intermediate value from the cipher text and the first key;

generating a second intermediate value from the first intermediate value and the second key; and

generating a plain-text envelope from the second intermediate value and the third key.

13. The method of Claim 12,

wherein the receiving further comprises receiving a first control number and

wherein the plain-text envelope comprises a plain-text message and a second control number.

14. The method of Claim 12, further comprising setting an integrity of the received cipher text to:

a "Can be Trusted" setting if the first control number is equal to the second control number and

a "Cannot be Trusted" setting if the first control number is not equal to the second control number.

15. The method of Claim 12, further comprising receiving a secret master key.

16. The method of Claim 12, wherein the generating the first key, the second key, and the third key comprises generating the first key, the second key, and the third key from the secret master key and the first control number.

17. The method of Claim 12, wherein the generating the first key, the second key, and the third key comprises generating the first key, the second key, and the third key from the secret master key and the first control number using a decryption function of a block cipher algorithm.

18. The method of Claim 12, wherein the generating the first intermediate value, the second intermediate value, and the plain-text envelope comprises using an decryption function of a block cipher algorithm.

19. The method of Claim 12, further comprising receiving the cipher text from a source over an unsecured medium.

20. An information handling system comprising:

one or more processors;

a memory accessible from the processors;

a receiver accessible from the processors for receiving data;

a message encryption tool that encrypts messages, the message encryption tool enabled to:

receive a plain text at the receiver;

generate a first key, a second key, and a third key;

generate a first intermediate value from the plain text and the first key;

generate a second intermediate value from the first intermediate value and the second key; and

generate a cipher text message from the second intermediate value and the third key.

21. The information handling system of Claim 20, wherein the message encryption tool is further enabled to receive a secret master key.

22. The information handling system of Claim 20, wherein the message encryption tool is further enabled to generate a first random number and a second random number.

23. The information handling system of Claim 20, wherein the message encryption tool is further enabled to generate a pad field such that a length of: the plain-text message, the first random number, the second random number, the pad field, and a number indicating a length of the pad field is an integer multiple of a block size of a block cipher algorithm.

24. The information handling system of Claim 20, wherein the message encryption tool is further enabled to construct a plain-text envelope comprising: the plain-text message, the first random number, the second random number, the pad field, and the number indicating the length of the pad field.

25. The information handling system of Claim 20, wherein, in order to generate the first intermediate value, the message encryption tool is further enabled to generate the first intermediate value from the plain-text envelope and the first key.

26. The information handling system of Claim 20, wherein the message encryption tool is further enabled to construct a protected-text envelope comprising: a number indicating a length of the protected-text envelope, the first random value, and the cipher-text message.

27. The information handling system of Claim 20, wherein the message encryption tool is further enabled to transmit the protected-text envelope to a target over an unsecured medium.

28. The information handling system of Claim 20, wherein, in order to generate the first key, the second key, and the third key, the message encryption tool is further enabled to generate the first key, the second key, and the third key from the secret master key and the first random number.

29. The information handling system of Claim 20, wherein, in order to generate the first key, the second key, and the third key, the message encryption tool is further enabled to generate the first key, the second key, and the third key from the secret master key and the first random number using a decryption function of a block cipher algorithm.

30. The information handling system of Claim 20, wherein, in order to generate the first intermediate value, the second intermediate value, and the cipher-text message, the message encryption tool is further enabled to use an encryption function of a block cipher algorithm.

31. An information handling system comprising:

one or more processors;

a memory accessible from the processors;

a receiver accessible from the processors for receiving data;

a message decryption tool that decrypts messages, the message decryption tool enabled to:

receive a cipher text at the receiver;

generate a first key, a second key, and a third key;

generate a first intermediate value from the cipher text and the first key;

generate a second intermediate value from the first intermediate value and the second key; and

generate a plain-text envelope from the second intermediate value and the third key.

32. The information handling system of Claim 31:

wherein, in order to receive the cipher text, the message decryption tool is further enabled to receive a first control number; and

wherein the plain-text envelope comprises a plain-text message and a second control number.

33. The information handling system of Claim 31, wherein the message decryption tool is further enabled to set an integrity of the received cipher text to:

a "Can be Trusted" setting if the first control number is equal to the second control number; and

a "Cannot be Trusted" setting if the first control number is not equal to the second control number.

34. The information handling system of Claim 31, wherein the message decryption tool is further enabled to receive a secret master key.

35. The information handling system of Claim 31, wherein, in order to generate the first key, the second key, and the third key, the message decryption tool is further enabled to generate the first key, the second key, and the third key from the secret master key and the first control number.

36. The information handling system of Claim 31, wherein, in order to generate the first key, the second key, and the third key, the message decryption tool is further enabled to generate the first key, the second key, and the third key from the secret master key and the first control number using a decryption function of a block cipher algorithm.

37. The information handling system of Claim 31, wherein, in order to generate the first intermediate value, the second intermediate value, and the plain-text envelope, the message decryption tool is further enabled to use an decryption function of a block cipher algorithm.

38. The information handling system of Claim 31, wherein the message decryption tool is further enabled to receive the cipher text from a source over an unsecured medium.

39. A computer program product stored on a computer operable media for encrypting a message, the computer program product comprising:

means for receiving a plain text;

means for generating a first key, a second key, and a third key;

means for generating a first intermediate value from the plain text and the first key;

means for generating a second intermediate value from the first intermediate value and the second key; and

means for generating a cipher text message from the second intermediate value and the third key.

40. The computer program product of Claim 39, further comprising means for receiving a secret master key.

41. The computer program product of Claim 39, further comprising means for generating a first random number and a second random number.

42. The computer program product of Claim 39, further comprising means for generating a pad field such that a length of: the plain-text message, the first random number, the second random number, the pad field, and a number indicating a length of the pad field is an integer multiple of a block size of a block cipher algorithm.

43. The computer program product of Claim 39, further comprising means for constructing a plain-text envelope, the plain-text envelope comprising: the plain-text message, the first random number, the second random number, the pad field, and the number indicating the length of the pad field.

44. The computer program product of Claim 39, wherein the means for generating the first intermediate value comprises means for generating the first intermediate value from the plain-text envelope and the first key.

45. The computer program product of Claim 39, further comprising means for constructing a protected-text envelope, the protected-text envelope comprising: a number indicating a length of the protected-text envelope, the first random value, and the cipher-text message.

46. The computer program product of Claim 39, further comprising means for transmitting the protected-text envelope to a target over an unsecured medium.

47. The computer program product of Claim 39, wherein the means for generating the first key, the second key, and the third key comprises means for generating the first key, the second key, and the third key from the secret master key and the first random number.

48. The computer program product of Claim 39, wherein the means for generating the first key, the second key, and the third key comprises means for generating the first key, the second key, and the third key from the secret master key and the first random number using a decryption function of a block cipher algorithm.

49. The computer program product of Claim 39, wherein the means for generating the first intermediate value, the second intermediate value, and the cipher-text message comprises means for using an encryption function of a block cipher algorithm.

50. A computer program product for decrypting a cipher text message, the computer program product comprising:

means for receiving a cipher text;

means for generating a first key, a second key, and a third key;

means for generating a first intermediate value from the cipher text and the first key;

means for generating a second intermediate value from the first intermediate value and the second key; and

means for generating a plain-text envelope from the second intermediate value and the third key.

51. The computer program product of Claim 50,

wherein the receiving further comprises means for receiving a first control number and

wherein the plain-text envelope comprises a plain-text message and a second control number.

52. The computer program product of Claim 50, further comprising means for setting an integrity of the received cipher text to:

a “Can be Trusted” setting if the first control number is equal to the second control number and

a “Cannot be Trusted” setting if the first control number is not equal to the second control number.

53. The computer program product of Claim 50, further comprising means for receiving a secret master key.

54. The computer program product of Claim 50, wherein the means for generating the first key, the second key, and the third key comprises means for generating the first key, the second key, and the third key from the secret master key and the first control number.

55. The computer program product of Claim 50, wherein the means for generating the first key, the second key, and the third key comprises means for generating the first key, the second key, and the third key from the secret master key and the first control number using a decryption function of a block cipher algorithm.

56. The computer program product of Claim 50, wherein the means for generating the first intermediate value, the second intermediate value, and the plain-text envelope comprises using an decryption function of a block cipher algorithm.

57. The computer program product of Claim 50, further comprising means for receiving the cipher text from a source over an unsecured medium.

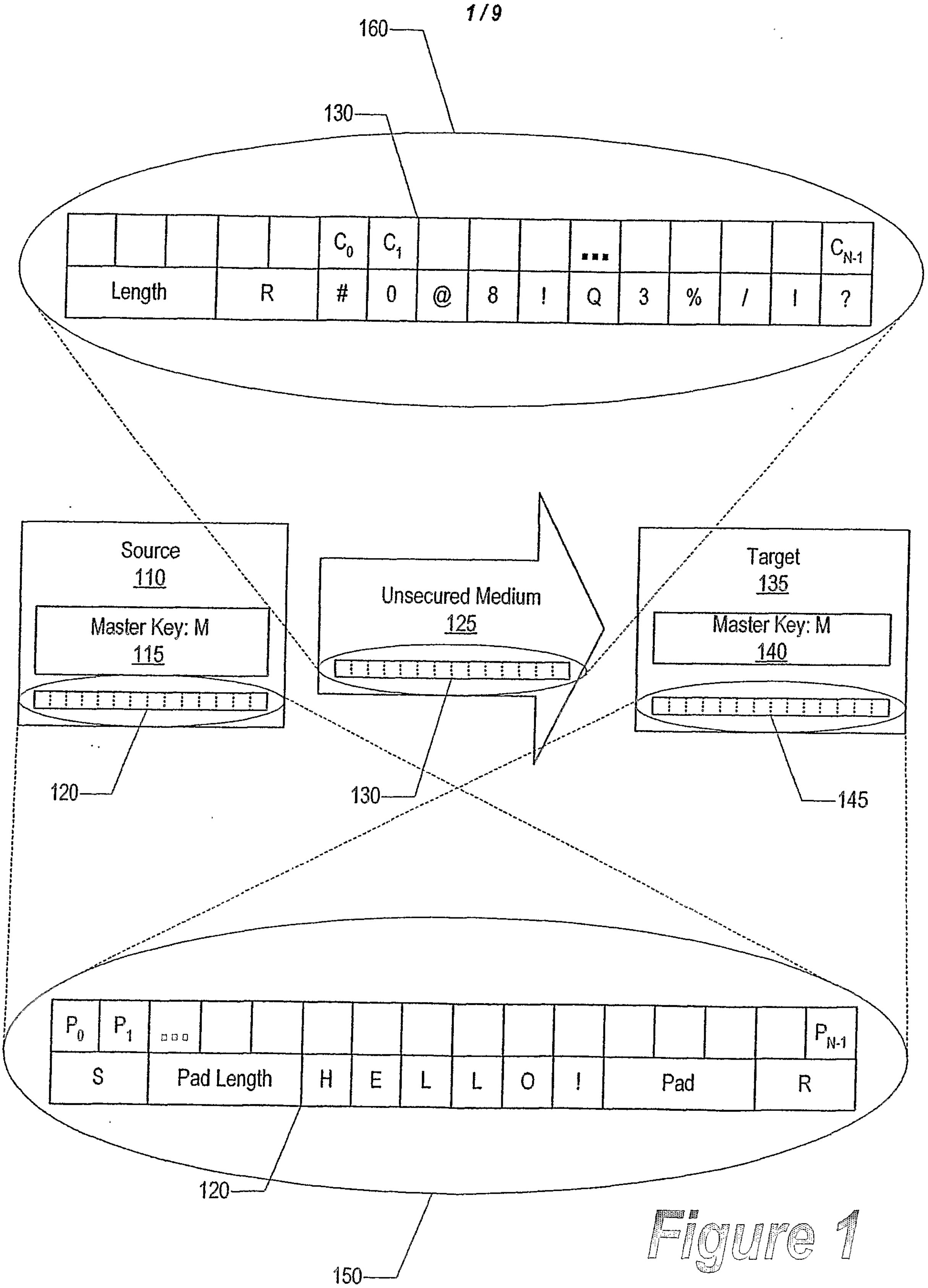

160
130
C0
C1
...
CN-1
Length
R
#
0
@
8
!
Q
3
%
/
|
?
Source
110
Master Key: M
115
120
Unsecured Medium
125
130
Target
135
Master Key: M
140
145
P0
P1
...
PN-1
S
Pad Length
H
E
L
L
O
!
Pad
R
120
150

Figure 1

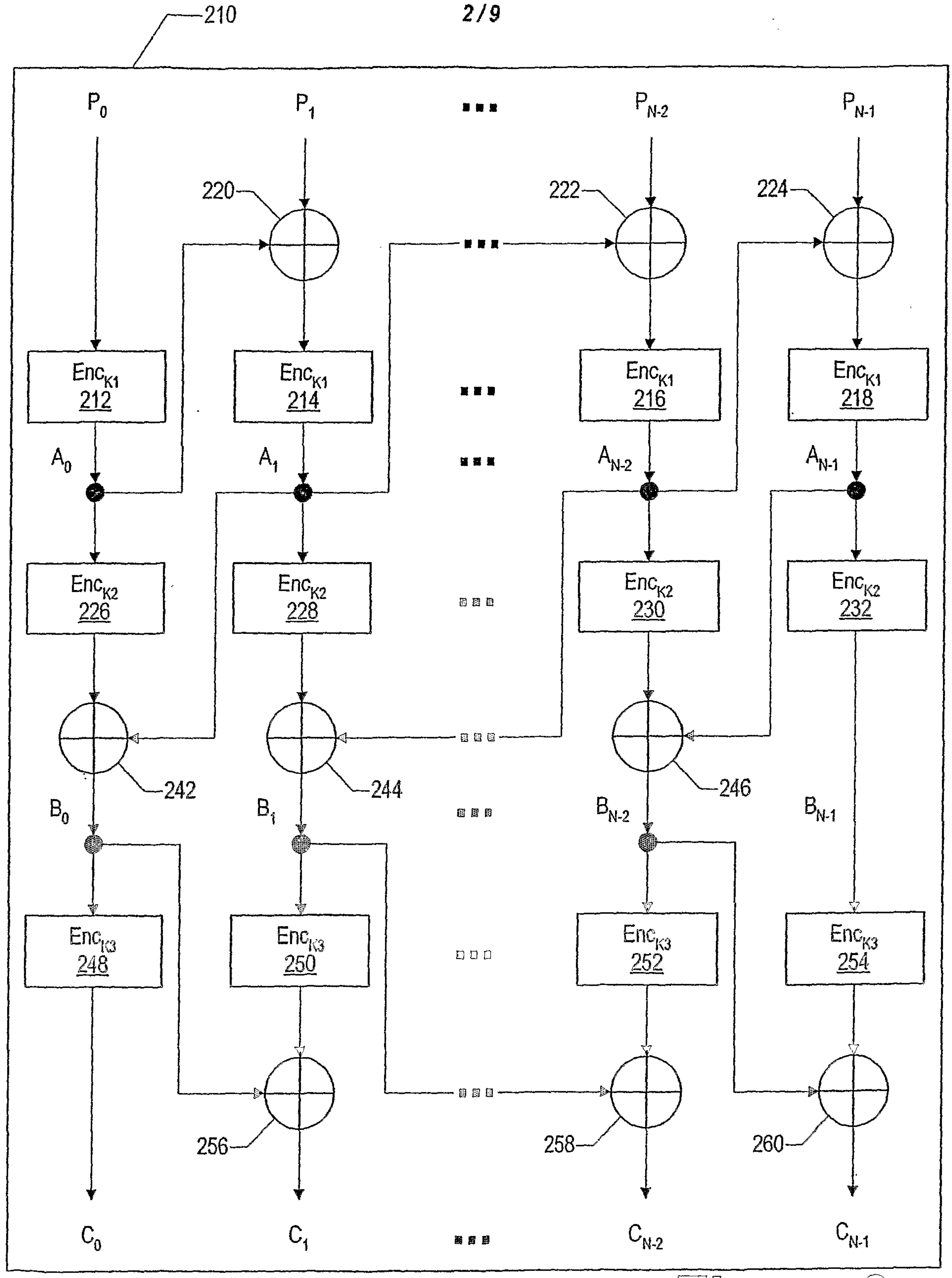

210
P0
P1
PN-2
PN-1
220
222
224
EncK1 212
EncK1 214
EncK1 216
EncK1 218
A0
A1
AN-2
AN-1
EncK2 226
EncK2 228
EncK2 230
EncK2 232
242
244
246
B0
B1
BN-2
BN-1
EncK3 248
EncK3 250
EncK3 252
EncK3 254
256
258
260
C0
C1
CN-2
CN-1

Figure 2

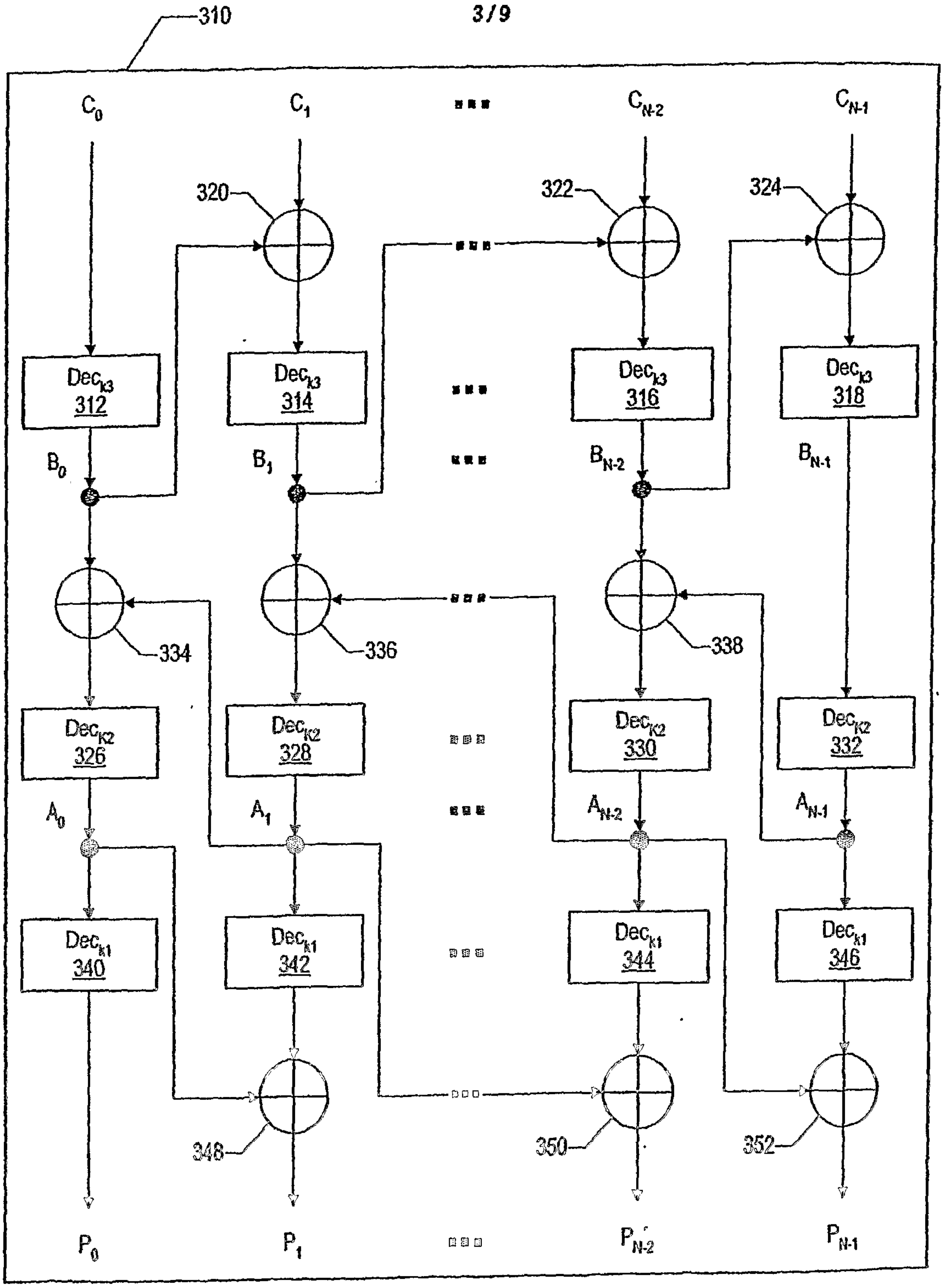

310
C_0
C_1
C_{N-2}
C_{N-1}
320
322
324
Dec_{k3} 312
Dec_{k3} 314
Dec_{k3} 316
Dec_{k3} 318
B_0
B_1
B_{N-2}
B_{N-1}
334
336
338
Dec_{K2} 326
Dec_{K2} 328
Dec_{K2} 330
Dec_{K2} 332
A_0
A_1
A_{N-2}
A_{N-1}
Dec_{k1} 340
Dec_{k1} 342
Dec_{k1} 344
Dec_{k1} 346
348
350
352
P_0
P_1
P_{N-2}
P_{N-1}

Figure 3

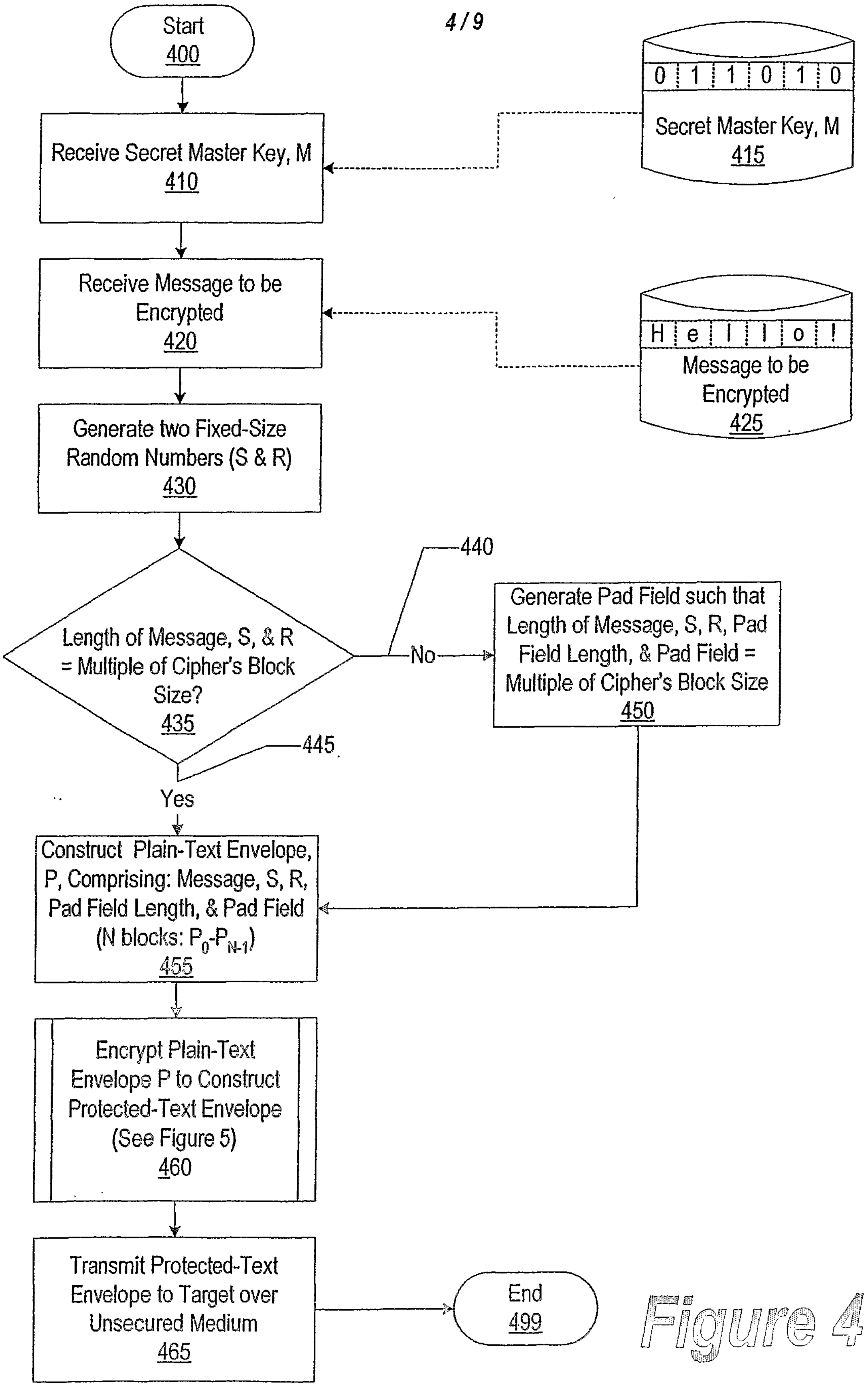

Start
400
Receive Secret Master Key, M
410
0 1 1 0 1 0
Secret Master Key, M
415
Receive Message to be Encrypted
420
H e l l o !
Message to be Encrypted
425
Generate two Fixed-Size Random Numbers (S & R)
430
440
Length of Message, S, & R = Multiple of Cipher's Block Size?
435
No
Generate Pad Field such that Length of Message, S, R, Pad Field Length, & Pad Field = Multiple of Cipher's Block Size
450
445
Yes
Construct Plain-Text Envelope, P, Comprising: Message, S, R, Pad Field Length, & Pad Field (N blocks: P0-PN-1)
455
Encrypt Plain-Text Envelope P to Construct Protected-Text Envelope (See Figure 5)
460
Transmit Protected-Text Envelope to Target over Unsecured Medium
465
End
499

Figure 4

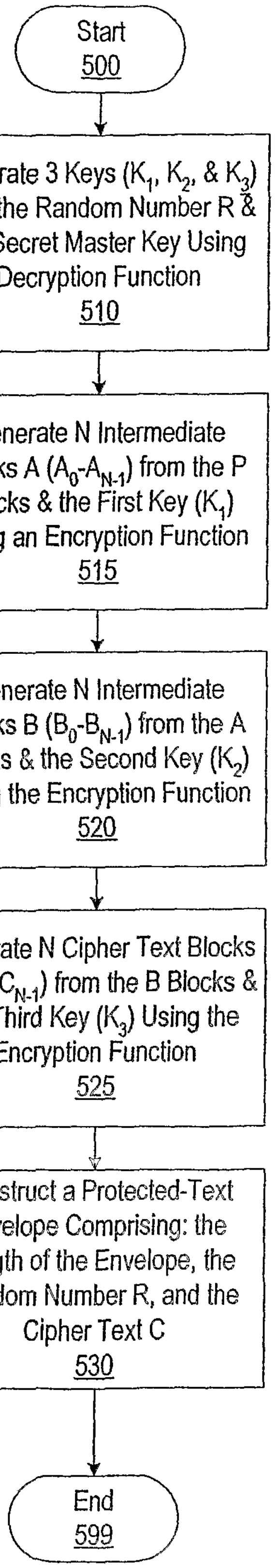

Start
500
Generate 3 Keys (K_1, K_2, & K_3) from the Random Number R & the Secret Master Key Using Decryption Function
510
Generate N Intermediate Blocks A (A_0-A_{N-1}) from the P Blocks & the First Key (K_1) Using an Encryption Function
515
Generate N Intermediate Blocks B (B_0-B_{N-1}) from the A Blocks & the Second Key (K_2) Using the Encryption Function
520
Generate N Cipher Text Blocks C (C_0-C_{N-1}) from the B Blocks & the Third Key (K_3) Using the Encryption Function
525
Construct a Protected-Text Envelope Comprising: the Length of the Envelope, the Random Number R, and the Cipher Text C
530
End
599

Figure 5

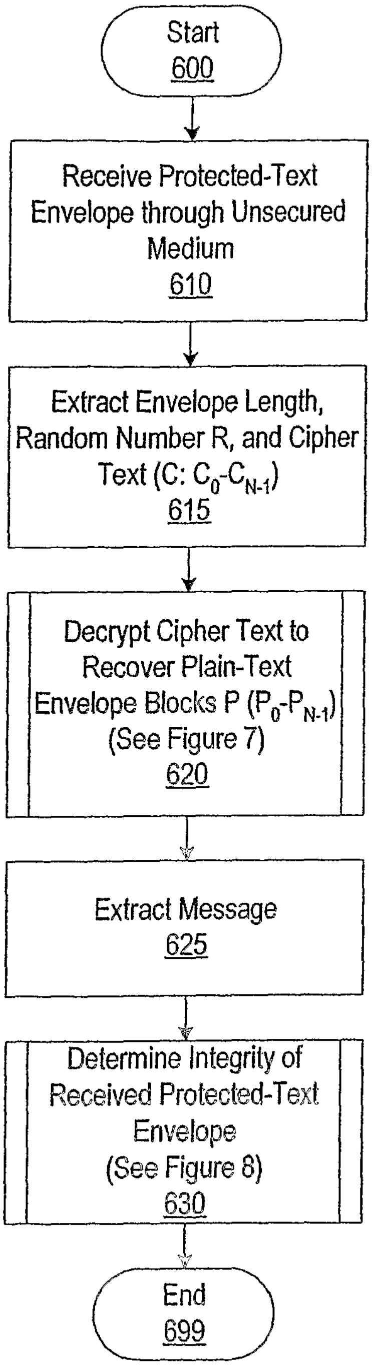

Start
600
Receive Protected-Text Envelope through Unsecured Medium
610
Extract Envelope Length, Random Number R, and Cipher Text (C: C_0-C_{N-1})
615
Decrypt Cipher Text to Recover Plain-Text Envelope Blocks P (P_0-P_{N-1}) (See Figure 7)
620
Extract Message
625
Determine Integrity of Received Protected-Text Envelope (See Figure 8)
630
End
699

Figure 6

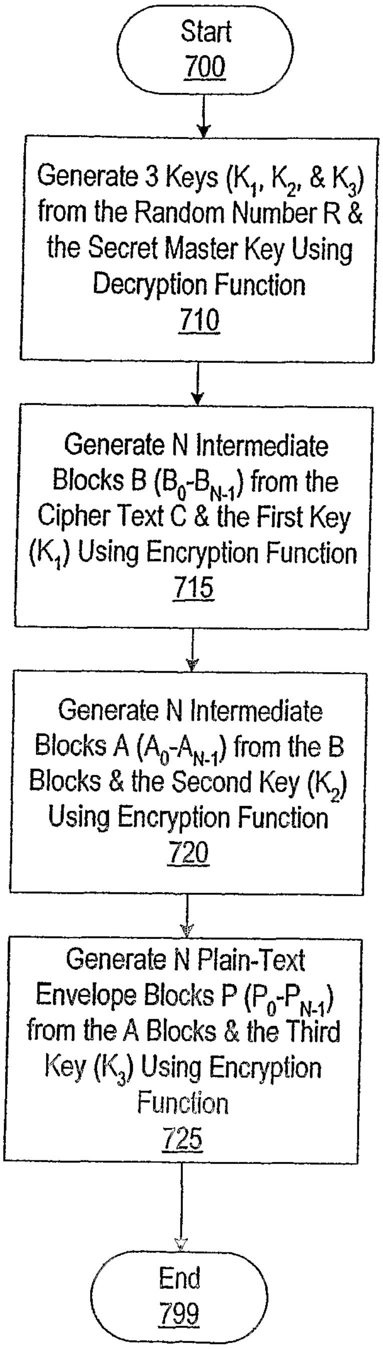

Start
700
Generate 3 Keys (K_1, K_2, & K_3) from the Random Number R & the Secret Master Key Using Decryption Function
710
Generate N Intermediate Blocks B (B_0-B_{N-1}) from the Cipher Text C & the First Key (K_1) Using Encryption Function
715
Generate N Intermediate Blocks A (A_0-A_{N-1}) from the B Blocks & the Second Key (K_2) Using Encryption Function
720
Generate N Plain-Text Envelope Blocks P (P_0-P_{N-1}) from the A Blocks & the Third Key (K_3) Using Encryption Function
725
End
799

Figure 7

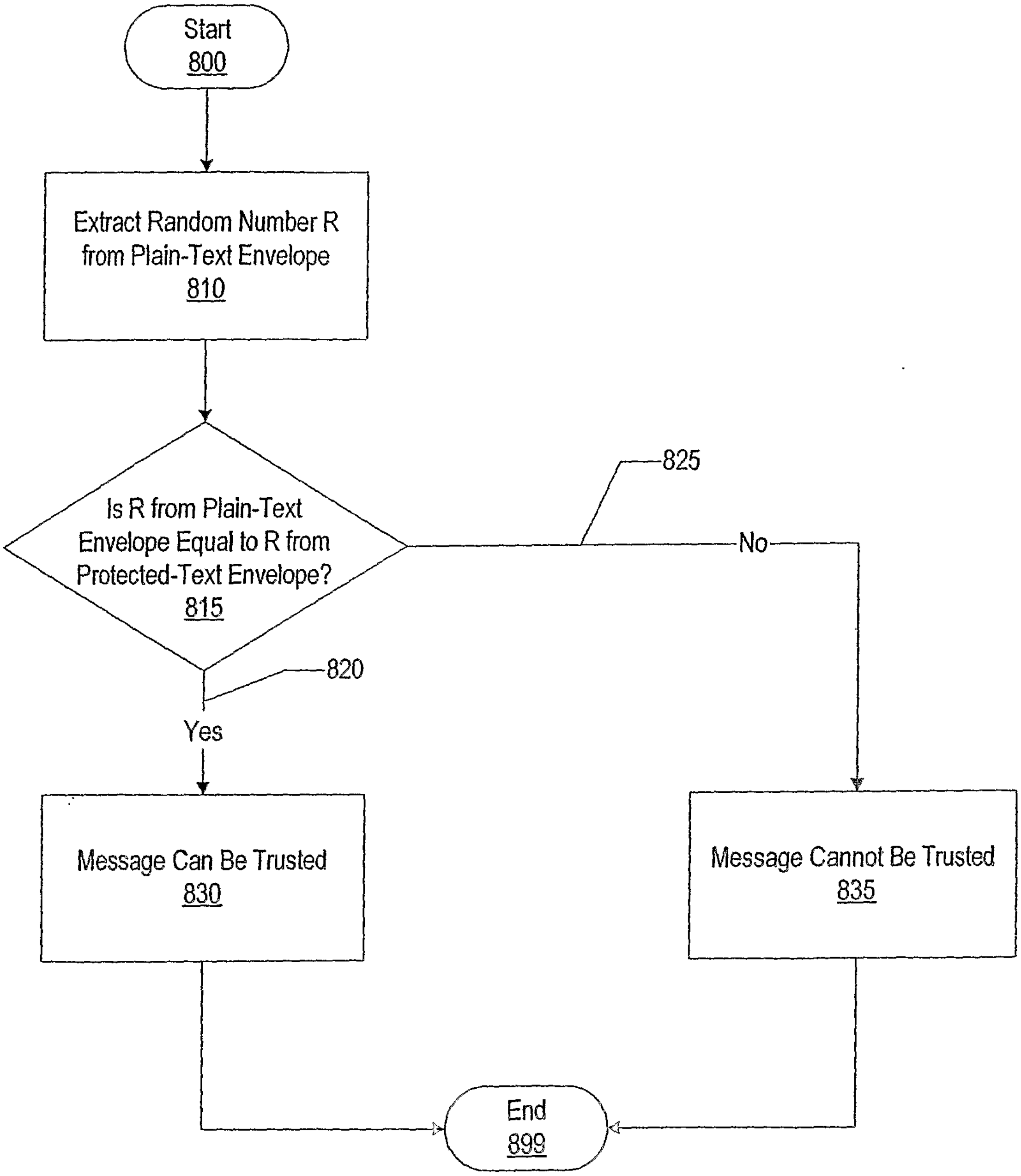

Start
800
Extract Random Number R from Plain-Text Envelope
810
Is R from Plain-Text Envelope Equal to R from Protected-Text Envelope?
815
825
No
820
Yes
Message Can Be Trusted
830
Message Cannot Be Trusted
835
End
899

Figure 8

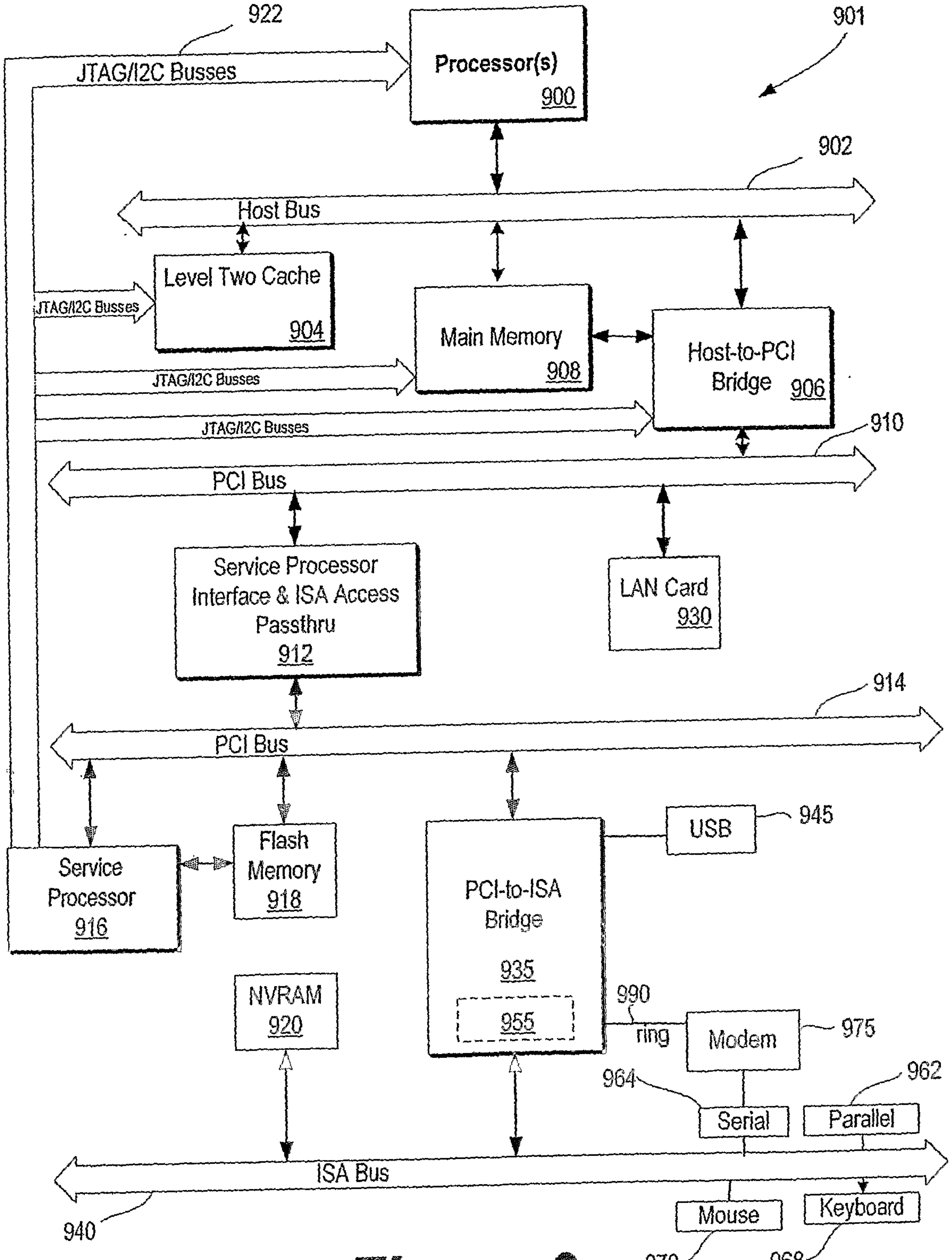

922
901
JTAG/I2C Busses
Processor(s)
900
902
Host Bus
Level Two Cache
904
JTAG/I2C Busses
Main Memory
908
JTAG/I2C Busses
Host-to-PCI
Bridge
906
JTAG/I2C Busses
910
PCI Bus
Service Processor
Interface & ISA Access
Passthru
912
LAN Card
930
914
PCI Bus
USB
945
Service
Processor
916
Flash
Memory
918
PCI-to-ISA
Bridge
935
955
990
ring
Modem
975
NVRAM
920
964
Serial
962
Parallel
ISA Bus
940
Mouse
Keyboard
970
968

Figure 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/33380

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : H04L 9/00
US CL : 380/277.000
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)
U.S. : 380/277.000, 28,29,42,44,45,46,255,259,283,284,286; 713/168,200,201

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPAT; US-PGPUB,

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Continuation Sheet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	SCHNEIER, BRUCE. Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., 1996. pp. 28-29, 169-175, 189-197, 357-363. see the entire document.	1, 6, 8, 11, 12, 18-20, 25, 27, 30, 31, 37-39, 44, 46, 49, 50 and 56-57 ---------- 3-5, 7, 9-10, 13-17, 21-24, 26, 28, 29, 32-36, 40-43, 45, 47, 48, and 51-55
Y,P	US 203/0159036 A1 (WALMSLEY et al) 21 August 2003 - see paragraphs 0498, 0499, 0692, see para. 0335.	3-5, 7, 13, 14, 22-24, 26, 32, 33, 41-43, 45 and 51-52

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* 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 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 27 September 2004 (27.09.2004)	Date of mailing of the international search report 03 DEC 2004
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. (703) 305-3230	Authorized officer Vincent Trans Peggy Harrod Telephone No. 703-305-3900

	PCT/US03/33380
INTERNATIONAL SEARCH REPORT	

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,517,567 A (EPSTEIN) 14 May 14, 1996 - see col. 7, lines 6-8, col. 7, lines 8-17 and col. 3 lines 39-49.	2, 9, 10, 15-17, 21, 28, 29, 34-36, 40, 47, 48 & 53 - 55
A	MENEZES, ALFRED, et al Handbook of Applied Cryptograph. CRC Press LLC, 1997. pp. 233-237, 271-173. see the entire document.	1 - 57
A	US 4,941,176 (MATYAS et al) 10 July 1990 - see the entire document.	1 - 57
A	US 4,941,176 (MATYAS et al) 10 July 1990 - see the entire document.	1 - 57
A	US 5,544,086 A (DAVIS et al) 06 August 1996 - see the entire document.	1 - 57
A	MERKLE, RALPH C., and Martin E. Hellman. "On the Security of Multiple Encryption". Communications of the ACM, Vol. 24, No. 7. July 1981. pp. 465-467. see the entire document.	1 - 57
A	COPPERSMITH, D. et al. "A proposed mode for triple-Des encryption". IBM Journal of Research and Development, Vol. 40, No. 2, March 1996. pp 253-262. see the entire document.	1 - 57

INTERNATIONAL SEARCH REPORT

PCT/US03/33380

Continuation of B. FIELDS SEARCHED Item 3:
brokenshire-daniel$,craft-david$,hofstee-ham$,multip$10,encrypt$3,trip$5,random pseudo$1,trip$5,three third multipl$10, integrity, authentic$5,key with generat$3,master.