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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 January 2002 (17.01.2002)

PCT

(10) International Publication Number
WO 02/05481 A1

(51) International Patent Classification[7]: H04L 9/00

(21) International Application Number: PCT/US01/02916

(22) International Filing Date: 30 January 2001 (30.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/610,723 6 July 2000 (06.07.2000) US

(71) Applicant and
(72) Inventor: LEE, Hitae [US/US]; CertandExodus Venture Inc., 4319 N. Larwin Avenue, Concord, CA 94521 (US).

(74) Agent: STANIFORD, Geoffrey, T.; Dergosits & Noah LLP, Four Embarcadero Center, Suite 1150, San Francisco, CA 94111 (US).

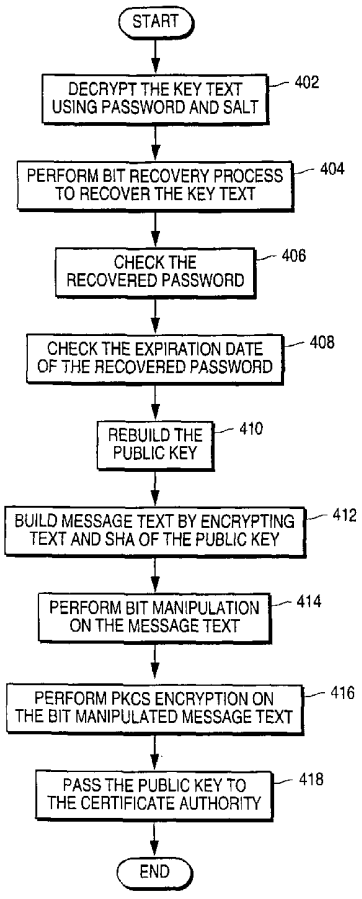
(81) Designated States *(national)*: AU, CA, CN, JP, KR, MX.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: THREE-WAY ENCRYPTION/DECRYPTION SYSTEM



(57) Abstract: A three-way encryption/decryption process for use in digital data transmission is described. A secret message is first encrypted using public key/private key encryption methods. Subsequent to public key encryption, the encrypted message text is bit manipulated to further code the secret message text. In one embodiment, the bit manipulation process includes a process of rearranging bits in a byte, thus preventing an attacker from deciphering any readable text even though an attacker might successfully decipher the encrypted message or acquire either the password and/or private key to decipher the encrypted message. A bit recovery process is used by the receiver after a public key decryption step to recover the bit manipulated encoded text.

WO 02/05481 A1

THREE-WAY ENCRYPTION/DECRYPTION SYSTEM

FIELD OF THE INVENTION

The present invention relates generally to computer networks, and more specifically, to an encryption and decryption process for the transmission of digital data.

BACKGROUND OF THE INVENTION

The Internet is basically a public network, but the messages sent over it often need to be kept private. Because of this, data encryption has become a fundamentally important aspect of data communication over the Internet and other public and private computer networks. Indeed, the success of the burgeoning electronic commerce industry relies on effective encryption means to protect sensitive data.

Traditional encryption methods rely on the concept of a key based cipher system to encode and decode transmitted data. A key is a piece of data, basically a long random number that can be used to encrypt or decrypt a given message. There are two basic type of encryption schemes, symmetric or single-key (one-way) encryption, and asymmetric or public key (two-way) encryption.

A symmetric-key cryptography system is an encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message. The key must be known at both ends of a connection, which poses a challenge with regard to communicating and protecting the integrity of the key. The problem with secret key cryptography, from the standpoint of transactions over the Internet, is that anyone who can get both the key and the encrypted

information can decrypt the information. Although symmetric-key systems are relatively simple and fast, their main drawback is that the two parties must somehow exchange the key in a secure way. The most popular symmetric-key system is the Data Encryption Standard (DES).

An asymmetric or public key encryption system is a cryptographic system that uses two keys, a public key known to everyone and a private known only to the recipient of the message. The sender uses the recipient's public key to encrypt the message. The recipient then uses his private key to decrypt it. Together the public and private keys make an 'asymmetric key pair'. Using this system, a message encrypted with a public key can only be decrypted with the matching private key, and vice versa. If a public key is used at encryption time, the message can only be unscrambled using the matching private key. This technique can safely be used to transmit a secret key, because only the intended recipient can decipher it. Public-key encryption represents an improvement over symmetric-key systems because the public key can be distributed in a non-secure way, and the private key is never transmitted.

A public-key scheme can also be used to show that a particular message is genuine. If a private key is used at encryption time, then the message can be read only with the corresponding public key. Without the private key, it is virtually impossible to forge an intelligible message that will unscramble with the public key, or to tamper successfully with a message that has already been encrypted.

An important characteristic of public key systems is that the public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them. One

difficulty with public-key systems is that the sender needs to know the recipient's public key to encrypt a message for the recipient. Because of this, public-key cryptography systems are not foolproof. One risk is that a sender must be certain of who owns a public key. It is usually necessary to have public keys verified by a trustworthy third party. Companies, such as VeriSign™, offer a commercial service for verifying and signing the public keys of other organizations.

An inherent disadvantage associated with in public-key schemes is that there is a fixed relationship between the public key and private key because one key unravels messages bound up by the other. Given the public key and knowledge of the encryption algorithm used, it can be possible to calculate the missing private key. In many cases, the calculation is very complex and time consuming, and given sufficiently long keys, public-key cryptography systems can be very difficult to break. However, breaking a public key system is not necessarily impossible.

Present electronic commerce ("e-commerce") applications rely on Certificate Authorities, which are trusted third-party organization or company to issue digital certificates used to create digital signatures and public-private key pairs. The role of the Certificate Authority in this process is to guarantee that the individual granted the unique certificate is the proper individual. To accomplish this, the Certificate Authority usually has an arrangement with a financial institution, such as a credit card company, to obtain information to confirm an individual's claimed identity. Certificate Authorities are a critical component in data security and electronic commerce because they guarantee that the two parties exchanging information are really who they claim to be. The most widely used standard for digital certificates is the X.509 standard. One significant disadvantage of present key distribution and

certificate processes is that keys are recycled. This creates a situation in which breaking one key results in the compromise of many other keys.

Thus, although present cryptography systems are effective in providing a reasonable level of security for most e-commerce transactions, the inherent limitations of current one-way and two-way systems are not impervious to attack. Given enough resources, a determined attacker can still break the standard cryptography systems currently in place.

SUMMARY AND OBJECTS OF THE INVENTION

It is an object of embodiments of the present invention to provide a secure method for the exchange of private and confidential financial transaction information over the Internet for the purpose of conducting business between a consumer and a business entity.

It is a further object of embodiments of the present invention to provide alternative levels of encryption and decryption based on the sensitive nature of financial transactions.

It is yet a further object of embodiments of the present invention to provide various delivery mechanisms for the distribution and housing of key information through the use of chip cards or proprietary client computer software.

A three-way encryption/decryption process for use in digital data transmission is described. In one embodiment, a message is encrypted using public key/private key encryption methods. Subsequent to public key encryption, the encrypted message text is bit manipulated to further code the secret message text. In one embodiment, the bit manipulation process includes a process of rearranging bits in a byte, thus preventing an attacker from deciphering any readable text even though an attacker might successfully decipher the encrypted message or acquire either the password and/or private key to decipher the encrypted message. A bit recovery process is used by the receiver after a public key decryption step to recover the bit manipulated encoded text.

Other objects, features, and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements, and in which:

Figure 1 illustrates a block diagram of a computer network system that implements embodiments of the present invention;

Figure 2 is a flowchart that illustrates the steps of performing digital data transmission using a three-way encryption/decryption process, according to one embodiment of the present invention;

Figure 3 is a flowchart that illustrates the steps of distributing a key using a three way encryption/decryption process according to one embodiment of the present invention;

Figure 4 is a flow chart that illustrates the step of encrypting a message using a three way encryption/decryption process according to one embodiment of the present invention;

Figure 5 is a flow chart that illustrates the step of decrypting a message using a three way encryption/decryption process according to one embodiment of the present invention;

Figure 6 is a flow diagram that illustrates a three-way encryption/decryption method for a key distribution process implemented in an Internet-based e-commerce transaction system, according to one embodiment of the present invention;

Figure 7 is a flow diagram that illustrates a three-way encryption/decryption method for an encryption process implemented in an Internet-based e-commerce transaction system, according to one embodiment of the present invention; and

Figure 8 is a flowchart that illustrates a three-way encryption/decryption method for a decryption process implemented in an Internet-based e-commerce transaction system, according to one embodiment of the present invention.

5

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A three-way encryption/decryption system for digital data transmission is described. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one of ordinary skill in the art, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form to facilitate explanation. The description of preferred embodiments is not intended to limit the scope of the claims appended hereto.

Hardware Overview

Aspects of the present invention may be implemented on one or more computers executing software instructions. According to one embodiment of the present invention, server and client computer systems transmit and receive data over a computer network or standard telephone line. The steps of accessing, downloading, and manipulating the data, as well as other aspects of the present invention are implemented by central processing units (CPU) in the server and client computers executing sequences of instructions stored in a memory. The memory may be a random access memory (RAM), read-only memory (ROM), a persistent store, such as a mass storage device, or any combination of these devices. Execution of the sequences of instructions causes the CPU to perform steps according to embodiments of the present invention.

The instructions may be loaded into the memory of the server or client computers from a storage device or from one or more other computer systems over a network connection. For example, a client computer may transmit a sequence of

instructions to the server computer in response to a message transmitted to the client over a network by the server. As the server receives the instructions over the network connection, it stores the instructions in memory. The server may store the instructions for later execution, or it may execute the instructions as they arrive over the network connection. In some cases, the downloaded instructions may be directly supported by the CPU. In other cases, the instructions may not be directly executable by the CPU, and may instead be executed by an interpreter that interprets the instructions. In other embodiments, hardwired circuitry may be used in place of, or in combination with, software instructions to implement the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software, nor to any particular source for the instructions executed by the server or client computers.

Figure 1 illustrates a computer network system 100 that implements one or more embodiments of the present invention. In system 100, a network server computer 104 is coupled, directly or indirectly, over line 125 to one or more network client computers 102 and 103 through a network 110. The network interface between server computer 104 and client computer 102 may also include one or more routers that serve to buffer and route the data transmitted between the server and client computers over lines 121 and/or 123. Network 110 may be the Internet, a Wide Area Network (WAN), a Local Area Network (LAN), or any combination thereof.

In one embodiment of the present invention, the server computer 104 is a World-Wide Web (WWW) server that stores data in the form of 'web pages' and transmits these pages as Hypertext Markup Language (HTML) files over the Internet network 110 to the client computer 102 as hidden field(s). For this embodiment, the

client computer 102 typically runs a "web browser" program to access the web pages served by server computer 104 and content provider 103.

In one embodiment of the present invention, server 104 in network system 100 is a server that executes an encryption/decryption software program 112. Client versions of the encryption/decryption software program 105 may also be executed on the client computers, such as client computer 102. The encryption/decryption program 112 may represent one or more executable program modules that are stored within network server 104 and executed locally within the server. Alternatively, however, it may be stored on a remote storage or processing device coupled to server 104 or network 110 and accessed by server 104 to be locally executed. In a further alternative embodiment of the present invention, encryption/decryption program 112 may be implemented in a plurality of different program modules, each of which may be executed by two or more distributed server computers coupled to each other, or to network 110 separately.

In one embodiment of the present invention, wherein network 110 is the Internet, network server 104 and content provider 103 execute a web server process (not shown to avoid obscuring the illustration) to provide HTML documents to client computers coupled to network 110. To access the HTML files provided by server 104, client computer 102 runs a web client process (typically a web browser, such as Netscape Navigator™ or Microsoft Explorer™) that accesses and provides links to web pages available on server 104 and other Internet server sites. It should be noted that a network system 100 that implements embodiments of the present invention may include a larger number of interconnected client and server computers than shown in

Figure 1. For this embodiment, the client computer 102 may access the Internet network 110 through an Internet Service Provider (ISP) 107.

As can be appreciated by those of ordinary skill in the art, the representative networked computers of Figure 1, such as network server computer 104 can be implemented as any standard computer that includes a CPU coupled through a bus to various other devices. These devices could include random access memory (RAM), a read only memory (ROM), and mass storage devices (e.g., a magnetic disk, optical compact disk, or tape drive for storing data and instructions). The computer also typically includes input/output devices, such as, a display device, keyboard, and network interface device, along with other similar devices or interfaces. Any of the computers in Figure 1 could be implemented in the form of personal computers, laptop computers, mainframe computers, or other type of workstation computers.

Encryption/Decryption Process

In one embodiment of the present invention, the client and server encryption/decryption software processes 112 and 105 serve to encrypt data transmitted between the network client 102 and the network server 104 and other networked computers, such as content providers on network 110. It is assumed that network client 102 implements known versions of a public key encryption system, such as the RSA Encryption system. In one embodiment, the encryption/decryption software processes 112 and 105 add a further level of encryption to the RSA system.

The encryption/decryption process described herein may be implemented to be run from the client computer 102 as a client side application 105 or applet. Preferably, however, the encryption/decryption process is executed from the server computer 104 as a server side program 112 or servlet. Alternatively, the

encryption/decryption process can consist of modules that reside on both the server and client computers.

In typical RSA Public Key Cryptography Systems (PKCS), a secret key is first generated using a one-way method, such as a DES method. The secret key is then used to encrypt the message, and the PKC key is used to encrypt the secret key. The PKC-encrypted secret key is then attached to the secret key-encrypted message. The standard RSA system thus includes two levels of encryption comprising encryption/decryption of the secret key (password) and the PKCS message encryption/decryption.

Embodiments of the present invention add a third level of encryption comprising a bit manipulation step prior to the PKCS encryption step, and a bit recovery step after the private key retrieval process. In one embodiment, the encryption/decryption process uses a simple methodology of rearranging bits in a byte, thus preventing an attacker from deciphering any readable text even though an attacker might successfully decipher the encrypted message, or acquire either the password and/or private key to decipher the encrypted message.

By providing different depths or levels of encryption, embodiments of the present invention enhance the security of private and confidential financial information transmitted over the Internet and increase throughput of the Certificate Authority engine where the validation of key information occurs. In typical e-commerce applications, the three-way encoding process of the present invention has several different applications.

For example, the bit manipulation step can be used when encoding a digital envelope with a user provided password or private key and revoking the

authentication certificate if it fails the password/private key decryption process prior to actual PKCS private key decryption. A digital envelope is a type of security that uses two layers of encryption to protect a message. First, the message itself is encoded using symmetric encryption, and then the key to decode the message is encrypted using public-key encryption.

In a second example, by arranging plain text in a predefined and secured mathematical layout that can only be recognized by all parties involved, the Certificate Authority can revoke the authentication certificate after a light decryption process such as DES or password based encryption prior to actual PKCS private key decryption.

In yet another example, by placing the Secure Hashing Algorithm (SHA) result of the public key as a part of the encrypted text, the Certificate Authority can validate the SHA value and revoke the certificate prior to the actual PKCS private key decryption. The SHA result refers to a 'hash' process that is used to generate a smaller digest of a message that depends on the contents of the large message, since, in practice, it is not necessary to scramble all of a large message when signing it. If the original is changed even slightly, the hash will generate a different result. The digest alone is encrypted using the private key, and sent out alongside the main message as a certificate. The recipient can apply the same hash algorithm to the incoming main message to create a new digest, decrypt the sender's certificate, and compare the two results. If the digests are identical then the code cannot have been altered and must have been sent by the owner of the appropriate public key.

Figure 2 is a high-level flowchart that illustrates the steps of a three-way encoding/decoding process, according to one embodiment of the present invention. In

general, the message recipient provides his or her public key to the message sender, the message sender then uses the public key to encrypt the message and sends the encrypted message to the recipient. The recipient then uses his or her private key to decrypt the sent message. Although the public key can be distributed to the message sender using known PKCS techniques, such as looking up the key on an Internet site or key registry or asking the recipient to provide his or her public key, for the embodiment illustrated in Figure 2, the public key is distributed to the user using a bit manipulation encryption process.

Thus, for the flowchart of Figure 2, in step 202, the public (PCKS) key for the recipient is distributed to the message sender. In general, this step entails retrieving a public key, performing a bit manipulation process on the key, encrypting the key, and then distributing the key to the sender. In step 204, the sender encrypts the message to be sent to the recipient. Step 204 generally entails decrypting the key and any password that may be used, rebuilding the public key, performing PKCS encryption, and certifying the key with a Certificate Authority. In step 206 the transmitted message is decrypted at the recipient's computer. This step generally entails the recipient retrieving his or her private key, performing PKCS decryption, checking the consistency of the secure hashing algorithm, and rebuilding the secret message. In accordance with embodiments of the present invention, during the message encryption and message decryption steps, 204 and 206, bit manipulation and bit recovery processes are executed to further encode the message being sent. The bit manipulation and bit recovery processes can also be used to encrypt the public key when it is distributed from the message recipient to the message sender. Each of the steps of flowchart of Figure 2 will be described in greater detail below.

If the three-way encryption process illustrated in Figure 2 is used in an e-commerce environment in which a transaction is processed between a buyer and a merchant site, payment processing of the transaction is performed in step 208. This step may entail various process specific steps, such as validating the buyers credit, validating the transaction, and fulfilling the order.

Figure 3 is a flowchart that illustrates the steps of distributing a public key from a message recipient to a message sender, using a three way encryption/decryption process according to one embodiment of the present invention. In step 302, the message recipient retrieves his or her public key. In step 304, a password that is used to code the public key for distribution to the message sender is then determined. Next, a salt value is determined using standard RSA salt routines. Salt is an additional string, usually 40 to 88 bits long, that can be added to a message as an additional measure to thwart attackers who try to precompute a large look-up table of possible encryption. The salt is appended to the encryption key, and the lengthened key is then used to encrypt the message.

In step 308, the key text is built by combining the recipient's public key, the sender's password, and the salt. In step 310 a bit manipulation process is performed on the key text. In one embodiment of the present invention, the bit manipulation process comprises a nibble exchange among pairs of nibbles that make up the message. Alternatively, the bit manipulation process can comprise a bit exchange process. The bit manipulation process is described in greater detail in the description that follows below. The bit manipulation process of step 310 serves to further scramble the key text so that an attacker cannot rebuild the key by breaking the password and salt values.

After the bit manipulation step of 310, the key text is encrypted with the password, step 312. The encrypted key is then distributed to the message sender, step 314. In one embodiment, the distribution step is accomplished by transmitting the encrypted key message to the recipient over the Internet using a secure electronic mail (e-mail) communication, or similar communication method.

Once the recipient's public key has been distributed to the sender, the sender encrypts the secret message to be sent to the recipient. Figure 4 is a flow chart that illustrates the steps of encrypting a message using a three way encryption/decryption process according to one embodiment of the present invention.

In step 402, the character string of the recipient's public key is decrypted using the password and the salt value. The key text is then recovered using a bit recovery process, step 404. In one embodiment, the bit recovery process is the opposite of the bit manipulation process performed in step 310. Thus, if the bit manipulation swapped the position of every two nibbles (four bits) comprising the original message, the bit recovery process would swap the nibbles back to their original position. In step 406, the recovered password is checked. If the password is approved, the expiration date of the recovered password is checked, step 408. In step 410, the recipient's public key is rebuilt.

Once the recipient's public key is rebuilt, the secret message is encrypted by the message sender. In step 412, the sender builds the message text by encrypting the text and the SHA of the public key. In step 414, a bit manipulation process is performed on the message text. The bit manipulation process performed in step 414 can be either a nibble exchange or a bit exchange of the ASCII characters that comprise the message. Such a process is described in the description that follows

below. In step 416, standard PKCS encryption is performed on the bit manipulated message text. This step is performed as many times as required for full encryption of the message. Multiple encryption steps might be required for messages that are longer than the maximum allowable length specified by present PKCS systems. In step 418, the public key is passed to a Certificate Authority to verify that the message recipient is a valid entity authorized to receive the message.

After the secret message has been encrypted, it is sent to the recipient for decryption. Figure 5 is a flow chart that illustrates the step of decrypting a message using a three way encryption/decryption process according to one embodiment of the present invention. In step 502, the message recipient retrieves his or her matching private key. A PKCS decryption process is then performed on the encrypted message, step 504. The decrypted message is now a bit manipulated version of the message, since a bit manipulation process was performed by the sender, step 506, as described in step 414 of Figure 4. To recover the message, the recipient performs a bit recovery process on the decrypted text. After the text is recovered, the consistency of the SHA value is checked, step 508. If the SHA value is consistent, the secret message is rebuilt, step 510.

Bit Manipulation Process

In one embodiment of the present invention, the three-way encryption/decryption method includes a bit manipulation/bit recovery scheme to further encrypt transmitted data. Such data could include keys, passwords, and the message data itself. For purposes of explanation, 'bit manipulation' refers to re-ordering the sequence of bits during encryption of a data string, and 'bit recovery' refers to recovering the original order of the sequence of bits during decryption of the

data string by performing the opposite sequence of the bit manipulation sequence. The bit manipulation and recovery steps are performed as part of the three-way encryption/decryption method as illustrated as part of the processes shown in Figures 3, 4, and 5.

Techniques for bit manipulation according to embodiments of the present invention include a nibble exchange, for example changing x34 to x43; or bit alteration, for example, change x34 = b0011 b0100 to xC1 = b1100 b0001, where 'x..' denotes a hexadecimal number and 'b..' denotes a binary number, in accordance with conventional usage.

For purposes of simplicity, nibble exchange examples will be illustrated, however, it should be noted that any form of bit alteration can be implemented for the process of three-way encryption/decryption described herein. As an example of a nibble exchange process, the following plain text sentence "This is the cat who ate the mouse!" is bit manipulated. The hexadecimal values of the ASCII format characters before nibble exchange are as follows:

54 68 69 73 20 69 73 20 74 68 65 20 63 61 69 20 77 68 6F 20 68 74 65 20 74 68 65 20 6D 6F 70 73 65 21

After a nibble exchange process, the values of the sentence are basically the inverted hexadecimal values as follows:

45 86 96 37 02 96 37 02 47 86 56 02 36 16 96 02 77 86 F6 02 86 47 56 02 47 86 56 02 D6 F6 07 37 56 12

The resulting value of the nibble exchange in ASCII format is as follows:

"E†–7_-7_G†V_6_-_w†ö_†GV_G†V_Öö_7V_"

Thus, it can be seen that the nibble exchange procedure effectively encrypts the original text message, and the actual nibble exchange scheme used by the sender must be known by the recipient in order to reconstruct the original message.

Key Distribution For Transaction Processing

In one embodiment of the present invention, the bit manipulation process is used as part of a key distribution process, as illustrated in the flowchart of Figure 3. Figure 6 is a flow diagram that illustrates a three-way encryption/decryption method for a key distribution process implemented in an Internet-based e-commerce transaction system, according to one embodiment of the present invention. In Figure 6, a client computer 602 executing a web browser program, such as Netscape Navigator™ or Microsoft Explorer™ establishes communication with an e-commerce site web server 608 through a local ISP 604 and e-commerce ISP 606. For a typical e-commerce transaction, the client computer 602 typically transmits sensitive information such as order and payment information. In system 600, the encryption/decryption process to provide the public key to the client 602 is provided by an encryption server 612, which is coupled to the e-commerce site server 608 through a private encryption server ISP 610.

An example of a bit manipulation process used to encrypt a public key in the transaction system 600 illustrated in Figure 6 is as follows. First, a proper public key or X.609 certificate based on floor level authorization is selected by the user of the client computer 602. For example, by choosing a 768 RSA public key, the public key will expressed in the following hexadecimal format:

E5 26 A1 E3 EF 61 71 2D EB 0C EB 4E 91 27 0F 8A 95 D7 FF 46 E8 7F 2D 2A FB E1 7F D8 0E E5 82 3B 22 3D E1 C3 1F 3C 85 CB BC 35 DE 11 61 28 C7 38 81 52 EF FE F7 0B 4C 22 5F BB 7D B6 0C 1F 3C 3C 40 C2 73 44 99 C4 81 72 C2 B9 3A EA 65 99 BC 6A 71 41 36 70 28 C4 C2 43 3B 88 21 E5 1D C8 83 67

For the transaction system illustrated in Figure 6, the password for the key is transmitted from the client computer 602 to the e-commerce web server 608 through the appropriate networking apparatus. The e-commerce web server 608 determines if a public key exists for the client computer user, step 621. If a public key does exist, that public key is used, and the process ends, since there is no need for further key distribution processing. If a public key for the user does not exist the password and transaction data is passed from the e-commerce web server 608 to the encryption server 612. The encryption server 612 then builds an input to the password based encryption process, step 622. The input to the password based encryption process in plain text is then constructed using the following formula to produce the secret message:

Key(byte[n]) + Nibble exchanged Password (byte[8]) + expiration date time stamp as CCCCYYDD (bytes(8)) + Key Length Type(byte(1)) + Key Sequence on database (byte[3]) + x'BB'(bytes[?])

In the above equation, ? is the number of x'BB's (padding characters) to make the length of the plain text to be a multiple of 8 bytes and if the resulting byte is 0x00, it will be changed to x'BB'.

For example, using the public key given above, a password 'redcoral', an expiration date of '20010101', a key length of '96' and a key sequence of '274', prior to nibble exchange produces the following hexadecimal format string:

E5 26 A1 E3 EF 61 71 2D EB 0C EB 4E 91 27 0F 8A 95 D7 FF 46 E8 7F 2D
2A FB E1 7F D8 0E E5 82 3B 22 3D E1 C3 1F 3C 85 CB BC 35 DE 11 61 28
C7 38 81 52 EF FE F7 0B 4C 22 5F BB 7D B6 0C 1F 3C 3C 40 C2 73 44 99
C4 81 72 C2 B9 3A EA 65 99 BC 6A 71 41 36 70 28 C4 C2 433B 88 21 E5
1D C8 83 67 72 65 64 63 6F 72 61 6C 32 30 30 31 30 31 30 31 60 BB 01 12
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB 00

In step 624, a nibble exchange process is performed on the key string.

Performing a nibble exchange process on the above key sequence produces the following constructed secret message for the public key:

5E 62 1A 3E FE 16 17 D2 BE C0 BE E4 19 72 F0 A8 59 7D FF 64 8E F7 D2
A2 BF 1E F7 8D E0 5E 28 B3 22 D3 1E 3C F1 C3 58 BC CB 53 ED 11 16 82
7C 83 18 25 FE EF 7F B0 C4 22 F5 BB D7 6B C0 F1 C3 C3 04 2C 37 44 99
4C 18 27 2C 9B A3 AE 56 99 CB A6 17 14 63 07 82 4C 2C 34 B3 88 12 5E
D1 8C 38 76 27 56 46 36 F6 27 16 C6 23 03 03 13 03 13 03 13 06 BB 10 21
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB 00

Prior to password based encryption, an eight byte randomly generated number hexadecimal number is generated as a salt value to be used as part of the password encryption process. For example, the salt value could be the string: A1 3E D0 69 67 37 BD B2.

In step 626, standard RSA encryption is performed on the key information using the salt. The resulting password-based encrypted value resulting from the combination of the constructed secret message shown above, plus the salt value exemplified immediately above, in hexadecimal format is shown as follows:

22 8B 46 A1 05 2E BE 35 11 DD A0 5E BA 78 3A D5 C5 75 F5 1D 77 2D
3B F1 98 FE 85 93 FC 33 35 BA 59 81 8A 8C 6C AF 24 61 F9 D5 55 A5 19
B4 93 AA 30 7E 27 B4 0B 72 74 EF 7C 7D 85 2B FE 3A 0F B6 B8 5B 64 72
2F 7A 66 BC 6F 5B 56 05 D9 43 6E 92 8E 04 45 CB 2E B9 54 B4 66 11 68
22 71 62 13 14 FA 2B 78 9B 1A 31 ED 73 CE D4 51 95 D7 7D F1 B2 44 58
FA 9B 70 18 4B A8 6A 40 95 35 CA C0 FD C4 79 BF DA AE 74 95 48 69 68
E7 49 23 87 91 74 6F

The above encrypted value and salt value can then be distributed to either a client computer desktop using Secure Sockets Layer, a Smart Card or an e-commerce

database repository 614. The Secure Sockets Layer (SSL) is a security-enhanced abstraction of sockets that provides transaction security at the link or transport level. With SSL, security properties are attached to the link or channel of communication between two parties, not the documents themselves.

In step 628 it is determined whether a smart card service is to be used to distribute the encrypted key to the client computer user. If the smart card service is to be used, the smart card is personalized to the user using the encrypted key and the salt value, step 630. The smart card is then mailed or otherwise delivered to the client computer user, step 632.

If a smart card is not used, as determined in step 628, the encrypted key is distributed to the user through the e-commerce site ISP 606 and e-commerce web server 608. The encrypted key is then ultimately delivered to the user through the user's local ISP 604 using SSL transmission processes.

Message Encryption For Transaction Processing

In one embodiment of the present invention, the three-way encryption/decryption process is used in an e-commerce application in which sensitive information, such as address and credit card information is transmitted from a buyer (message sender) to an e-commerce merchant web site (message recipient) over a network.

Figure 7 is a flow diagram that illustrates a three-way encryption/decryption method for an encryption process implemented in an Internet-based e-commerce transaction system, according to one embodiment of the present invention.

For the exemplary transaction illustrated as process 700 in Figure 7, the client computer user (buyer) 602 may pass information that comprises the secret message,

password, salt, and the encrypted public key to an e-commerce site web server through one or more ISP's 604 and 606. For example, in plain text readable format, the message could be as follows: 4211232155678123 HITAE LEE 4319 N. LARWIN AVE. CONCORD CA 94521 0912 0001200.34 WORLD MUSIC AND VIDEO EQUIPMENT 1223-3451-1 BANK OF AMERICA. In hexadecimal format, this message is represented as follows:

34 32 31 31 32 33 32 31 35 35 36 37 38 31 32 33 20 48 49 54 51 45 4C 45 45
34 33 31 39 20 4E 2E 4C 41 52 57 49 4E 20 41 56 2E 20 43 4F 4E 43 4F 52
44 20 43 41 20 39 34 35 32 31 20 30 39 31 32 20 30 30 30 31 32 30 30 2E 33
34 20 57 4F 53 4C 44 20 4D 55 53 49 43 20 41 4E 44 20 56 49 44 45 4F 20 45
51 55 49 50 4D 45 4E 54 20 31 32 32 33 2D 33 34 35 31 2D 31 20 42 41 4E
4B 20 4F 46 20 41 4D 45 42 49 43 41 BB

In one embodiment of the present invention, the secret message is encrypted using an encryption server, such as encryption server 612 of figure 6. The encryption server performs a series of operations illustrated as process steps within block 702. In step 704, the e-commerce site 608 decrypts the encrypted public key using the provided password and salt. Successful decryption requires the proper password and salt value. If the recipient, or a would be attacker supplies an incorrect password; for example, 'joel', the decryption process will generate the following value:

9E 07 C6 02 6D FC F3 EB 34 5F B7 4C 40 0E 28 72 77 89 5A D6 4C ED 1F
7B CD 89 F4 DB 69 FF 30 73 C6 B3 FE A4 FA 4F DD 75 6E BB ED 06 F0
12 4D CD 7D 1A 59 52 E9 09 3A CA 01 E7 8C 38 46 99 44 85 FA AA 29 32
DC BB 63 DC C4 0B DC 69 C9 03 24 30 C6 02 37 92 3E 84 3F 58 C9 CC 4B
A1 70 14 95 DF 8B 3A 1B 96 2F 5F 4D 75 83 44 2F 78 80 3F 42 C8 B8 CD
51 43 90 82 2B F9 58 EC 1A 2B 4F 11 E8 98 87 44 D5 C9 FD 72 0D C8

If a would be attacker tries to decipher this value by displaying the contents of this returned value it would display as a random selection of characters such as:

ž_Ć_müóë4_·L@^(rw‰ZÖLí {Í‰ôŰi·0sĆłţ¤úOÝun»í_đ_MÍ}_YR
:Ę_çŚ8F™D…ůŞ)2Ü»cÜÄÜiÉ_$0Ć_7'>„?XÉĚK˘p_•ß‹:_–
/_Mu□D/x??BČ,ÍQC□,+ůXě_+O_č□‡D ŐÉýrČ

The next step is to compute the offset of the password using the key length. Before a nibble exchange step, the hexadecimal value of the offset is: 8B 3A 1B 96 2F 5F 4D 75. After the nibble exchange step, the hexadecimal value of the offset is: B8 3A B1 69 F2 F5 D4 57, which is not 'joel', so the password is in fact incorrect.

If a recipient enters the correct password ('redcoral'), the decryption process will generate the following value:

5E 62 1A 3E FE 16 17 D2 BE C0 BE E4 19 72 F0 A8 59 7D FF 64 8E F7 D2
A2 BF 1E F7 8D E0 5E 28 B3 22 D3 1E 3C F1 C3 58 BC CB 53 ED 11 16 82
7C 83 18 25 FE EF 7F B0 C4 22 F5 BB D7 6B C0 F1 C3 C3 04 2C 37 44 99
4C 18 27 2C 9B A3 AE 56 99 CB A6 17 14 63 07 82 4C 2C 34 B3 88 12 5E
D1 8C 38 76 27 56 46 36 F6 27 16 C6 23 03 03 13 03 13 03 13 13 BB 10 21
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB 00

If a would be attacker tries to decipher this value by displaying the contents of this returned value it would display as the following string:

^b_>þ__Ò¾À¾ä_rð¨Y}ÿdŽ÷Ò¢¿-÷ à^(³"Ó-<ñÃX¼ËSí__,|ƒ_%þï▒°Ä"õ»×kÀñÃÃ
_,7D™L_',›£®V™Ë¦__c_,L,4³^_^ÑŒ8v'VF6ö'_Æ#________»_!»»»»»»»»»»»»»»»»»»»»»»»»
»»»»»

Thus, even though the returned deciphered value is correct, an attacker will not recognize or consider it as a decrypted text value since this value still needs to be passed through the nibble exchange process.

The next step is to compute the offset of the password using the key length and retrieve the password from the decrypted public key and perform the nibble exchange, step 706. For example, the retrieved password before the nibble exchange could be: 27 56 46 36 F6 27 16 C6. If a would be attacker tries to decipher this value by displaying the contents of this returned value it would display as: 'VF6ö'_Æ

The retrieved password after nibble exchange is as follows: 72 65 64 63 6F 72 61 6C. This produces the following string 'redcoral', which is the proper password.

In step 708 it is determined whether the password matches. If the retrieved password does not match, the transaction is declined, this produces an 'authorization failed' situation. If the password validation process fails more than a pre-determined maximum number of failures, as determined in step 710, the certificate is revoked and the registered owner is informed, such as through e-mail using the owner's registered e-mail address or through other validated means, step 712.

If the password matches, the server next checks whether the key is expired, step 714. If a key has expired or is revoked, the transaction is declined and the e-commerce site is informed and instructed to expel or terminate the shopping process, step 712.

If, in step 714, it is determined that the password has not expired, it is assumed that the password is valid, and the process continues from step 716. In step 716, a nibble exchange process is performed on the retrieved public key. An exemplary public key, after a nibble exchange process can be expressed as:

E5 26 A1 E3 EF 61 71 2D EB 0C EB 4E 91 27 0F 8A 95 D7 FF 46 E8 7F 2D
2A FB E1 7F D8 0E E5 82 3B 22 3D E1 C3 1F 3C 85 CB BC 35 DE 11 61 28
C7 38 81 52 EF FE F7 0B 4C 22 5F BB 7D B6 0C 1F 3C 3C 40 C2 73 44 99
C4 81 72 C2 B9 3A EA 65 99 BC 6A 71 41 36 70 28 C4 C2 43 3B 88 21 E5
1D C8 83 67

A hashing operation is then performed on the nibble-exchanged public key, step 718. For this step, SHA1 hashing is performed on the recovered public key to produce a 20-byte SHA1 hexadecimal value as follows: 4A F8 29 C9 55 43 C2 D0 83 43 DF 0D 32 D8 95 5D.

In step 720, the secret message to be sent from client computer user to the e-commerce site is built for PKCS encryption. This step entails bit manipulating (nibble exchanging) the hexadecimal characters comprising the ASCII text of the

message. For example, a nibble exchange for an exemplary secret message can be as shown:

43 23 13 13 23 33 23 13 53 53 63 73 83 13 23 33 02 84 94 45 15 54 C4 54 54
43 33 13 93 02 E4 E2 C4 14 25 75 94 E4 02 14 65 2E 02 34 F4 E4 34 F4 25
44 02 43 14 02 93 43 53 23 13 02 03 93 13 23 02 03 03 03 13 23 03 03 E2 33
43 02 75 F4 35 C4 44 02 D4 55 35 94 34 02 14 E4 44 02 65 94 44 54 F4 02 54
15 55 94 05 D4 54 E4 45 02 13 23 23 33 D2 33 43 53 13 D2 13 02 24 14 E4
B4 02 F4 64 02 14 D4 54 24 94 34 14 BB

In one embodiment of the present invention, certain encryption adjustments must be performed since RSA PKCS processes generally cannot encrypt a string longer than the length of the public key, that is 11 bytes. For the above example, the secret message is broken into two groups, one group of 64 bytes and the second group of 72 bytes. Standard PKCS encryption techniques are then performed on the nibble exchanged secret message. For a broken down message, such as that described above, PKCS encryption will be performed on each group. Thus, as shown in step 722, multiple PKCS processes are performed on the nibble exchanged secret message.

For our example, the first encryption produces the plain text string (in hexadecimal):

43 23 13 13 23 33 23 13 53 53 63 73 83 13 23 33 02 84 94 45 15 54 C4 54 54
43 33 13 93 02 E4 E2 C4 14 25 75 94 E4 02 14 65 2E 02 34 F4 E4 34 F4 25
44 02 43 14 02 93 43 53 23 13 02 03 93 13 23

The encrypted text for this string, in hexadecimal, is:

0F 1A 25 A3 FA 07 7D D8 C7 05 EA BA 28 D5 5E CB 98 D8 0B 5D 98 FC
2D 0D A7 D8 48 E3 AC DA BF C4 DF D1 C3 B4 6D A3 BB A4 BA 2C 7E
45 1C A1 16 1F 87 4E 69 D5 6B A0 E4 42 94 70 98 CD 6A BF DC A9 4F F5
C8 80 D5 24 D7 33 CA 3B 56 C2 49 A3 04 F9 81 D1 EA 24 56 DB A2 24 51
50 C2 8B 57 BD 93 D8

The second encryption operation (for the second group) produces the plain text (in hexadecimal):

02 03 03 03 13 23 03 03 E2 33 43 02 75 F4 35 C4 44 02 D4 55 35 94 34 02 14 E4 44 02 65 94 44 54 F4 02 54 15 55 94 05 D4 54 E4 45 02 13 23 23 33 D2 33 43 53 13 D2 13 02 24 14 E4 B4 02 F4 64 02 14 D4 54 24 94 34 14 BB

The encrypted text for this string, in hexadecimal, is:

9E C2 73 DF 0B D3 7D D7 01 75 B0 19 5B 78 61 E3 21 12 87 A1 3F 8D 9C ED CE A1 ED D7 DC 7A 3C 02 B9 CB 53 8F 8F 93 29 2E E3 63 E8 61 21 61 16 C0 6B 74 C1 45 AE 85 29 31 B9 65 C2 04 5A D2 B0 82 B6 2F 93 41 15 A1 30 7D 24 51 FE 9C C3 F7 C0 EE F0 76 19 0A 2F 24 0B 92 5E 86 73 82 A1 4D 8F 14

In step 724, the encrypted secret message is built using the SHA1 of the public key and the encrypted message text. In one embodiment, the encrypted message is built using the following equation:

First block of Encrypted plain text + SHA1 of public key + Second block of Encrypted plain text + length of the first secret message block + key length(byte[2]) + key sequence(byte[4])

The encrypted message is then passed to a Certificate Authority for validation, step 726.

Message Decryption For Transaction Processing

In one embodiment of the present invention, decryption of the encrypted message using the three-way encryption/decryption process for an e-commerce application is performed by a Certificate Authority.

Figure 8 is a flowchart that illustrates a three-way encryption/decryption method for a decryption process implemented in an Internet-based e-commerce transaction system, according to one embodiment of the present invention. In step 802, the Certificate Authority retrieves the key length and key sequence from the encrypted message. In step 804, the Certificate Authority retrieves the matching public key from the Certificate Authority repository using the key length and key sequence. The Certificate Authority performs hashing (SHA1) on the retrieved public

key. In the above example message, an SHA1 value is placed after first block of the encrypted text which starts at 97 to 106. An example of an SHA1 string for a particular secret message is: 4A F8 29 C9 55 43 C2 D0 83 43 DF 0D 32 D8 95 5D 85 73 DC 14 BB BB BB BB

The Certificate Authority next retrieves the SHA1 from the secret message and validates the public key/private key pair, step 806. If, in step 808, it is determined that the recovered SHA1 and resulting SHA1 of the retrieved public key do not match, the transaction/certificate is revoked, and a rejection is sent to the e-commerce site, step 828.

If the SHA1 values do match, the encrypted secret message information (called signed application data) blocks are retrieved by joining the blocks of the message together, step 810. For the message example provided above, the first and second part of the encrypted message from bytes 1 to 96 and bytes 121 to 216 are joined to retrieve the message. Next, the proper RSA PKCS private key is retrieved using key length and key sequence information. The RSA PKCS private key has the following components: Public Key Modulus, Public Exponent, Private Exponent, Primary Modulus 1, Primary Modulus 2, Primary Exponent 1, Primary Exponent 2, and Coefficient. In step 812 PKCS decryption is performed on the private key. Again, if the string is too long, repeated decryption on the encrypted message may be performed. In step 814, it is determined whether all of the blocks have been retrieved. If not, the process loops back from step 810. Once all the message blocks have been decrypted, the decrypted blocks are combined, step 816. In step 818, a nibble exchange process is executed on the combined message blocks to rebuild the original

secret message. In step 820, the original secret message is rebuilt, and the message is then sent to the e-commerce site, step 820.

At this point, the further processing of the secret message conforms to the requirements of the system in which the e-commerce transaction is being performed. For example, if the transaction is being performed using a credit system, certain validation steps may need to be implemented using a credit bureau and/or other financial institutions. Ultimately, the secret message is transmitted to the e-commerce site for fulfillment of the transaction.

In one embodiment of the present invention, the transmission client/server software is implemented in a network that utilizes point to point telephony infrastructure. It is to be noted, however, that alternative embodiments can be implemented for use with point to point video conferencing or point to multi-point homing, or similar network systems. In addition, embodiments of the present invention can be implemented over networks that partially or wholly implement wireless data communication technology.

Although some of the Figures and associated description are largely directed to embodiments that utilize technology that is specific to the Internet and the World Wide Web, it should be noted that embodiments of the present invention can also be used in the context of other networked computer systems, such as local area networks, wide area networks, and other proprietary networks.

In the foregoing, a system has been described for increasing the security of messages transmitted over a network through a three-way encryption/decryption method. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes

may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method of transmitting data from a first computer to a second computer, comprising the steps of:

receiving a public key for a user of the second computer;

obtaining a password for the first user;

encrypting the public key using asymmetric encryption;

rearranging the order of the bits that comprise the encrypted public key;

encrypting the rearranged encrypted public key with the password; and

distributing the encrypted key text to the user of the first computer.

2. The method of claim 1 further comprising the steps of:

decrypting the encrypted key text using the password;

recovering the key text by reversing the step of rearranging the order of the bits that comprise the encrypted public key;

encrypting the message comprising the data to be transmitted from the first computer to the second computer;

rearranging the order of the bits that comprise the message; and

encrypting the message using asymmetric encryption.

3. The method of claim 2 further comprising the steps of:

retrieving a private key corresponding to the public key of the second user;

performing an asymmetric decryption process on the encrypted message; and

recovering the message data by reversing the step of rearranging the order of the bits that comprise the encrypted message.

4. The method of claim 1 wherein the step of rearranging the order of the bits that comprise the encrypted public key comprises the step of switching the position of each pair of nibbles comprising the hexadecimal representation of the key.

5. The method of claim 2 wherein the step of rearranging the order of the bits that comprise the message comprises the step of switching the position of each pair of nibbles comprising the hexadecimal representation of the message.

6. The method of claim 1 wherein the step of rearranging the order of the bits that comprise the encrypted public key comprises the step of switching the position of each pair of bits comprising the binary representation of the key.

7. The method of claim 2 wherein the step of rearranging the order of the bits that comprise the message comprises the step of switching the position of each pair of bits comprising the binary representation of the message.

8. The method of claim 3 wherein the first computer and the second computer are coupled over a network.

9. The method of claim 8 wherein the network comprises the Internet, and wherein the first computer and second computer respectively execute server and client processes operable to transmit and receive data files over the World Wide Web portion of the Internet, and further wherein the document data comprises Hypertext Markup Language (HTML) data executable by the first computer and second computer

10. A system for transmitting secret message data from a first computer to a second computer, comprising one or more distributed computers configured to:

receive a public key for a user of the second computer;

obtain a password for the first user;

encrypt the public key using asymmetric encryption;

rearrange the order of the bits that comprise the encrypted public key;

encrypt the rearranged encrypted public key with the password; and

distribute the encrypted key text to the user of the first computer.

11. The system of claim 10 wherein the one or more distributed computers are further configured to:

decrypt the encrypted key text using the password;

recover the key text by reversing the step of rearranging the order of the bits that comprise the encrypted public key;

encrypt the message comprising the data to be transmitted from the first computer to the second computer;

rearrange the order of the bits that comprise the message; and

encrypt the message using asymmetric encryption.

12. The system of claim 11 wherein the one or more distributed computers are further configured to:

retrieve a private key corresponding to the public key of the second user;

perform an asymmetric decryption process on the encrypted message; and

recover the message data by reversing the step of rearranging the order of the bits that comprise the encrypted message.

13. The system of claim 10 wherein the one or more distributed computers are further configured to rearrange the order of the bits that comprise the encrypted public key by switching the position of each pair of nibbles comprising the hexadecimal representation of the key.

14. The system of claim 11 wherein the one or more distributed computers are further configured to rearrange the order of the bits that comprise the encrypted public key by switching the position of each pair of nibbles comprising the hexadecimal representation of the message.

15. The system of claim 10 wherein the one or more distributed computers are further configured to rearrange the order of the bits that comprise the encrypted public key by switching the position of each pair of bits comprising the binary representation of the key.

12. The system of claim 11 wherein the one or more distributed computers are further configured to:

retrieve a private key corresponding to the public key of the second user;

perform an asymmetric decryption process on the encrypted message; and

recover the message data by reversing the step of rearranging the order of the bits that comprise the encrypted message.

13. The system of claim 10 wherein the one or more distributed computers are further configured to rearrange the order of the bits that comprise the encrypted public key by switching the position of each pair of nibbles comprising the hexadecimal representation of the key.

14. The system of claim 11 wherein the one or more distributed computers are further configured to rearrange the order of the bits that comprise the encrypted public key by switching the position of each pair of nibbles comprising the hexadecimal representation of the message.

15. The system of claim 10 wherein the one or more distributed computers are further configured to rearrange the order of the bits that comprise the encrypted public key by switching the position of each pair of bits comprising the binary representation of the key.

16. The system of claim 11 wherein the one or more distributed computers are further configured to rearrange the order of the bits that comprise the encrypted public

key by switching the position of each pair of bits comprising the binary representation of the message.

17. The system of claim 12 wherein the one or more distributed computers comprise a first computer and the second computer coupled over a network.

18. The system of claim 17 wherein the network comprises the Internet, and wherein the first computer and second computer respectively execute server and client processes operable to transmit and receive data files over the World Wide Web portion of the Internet, and further wherein the document data comprises Hypertext Markup Language (HTML) data executable by the first computer and second computer.

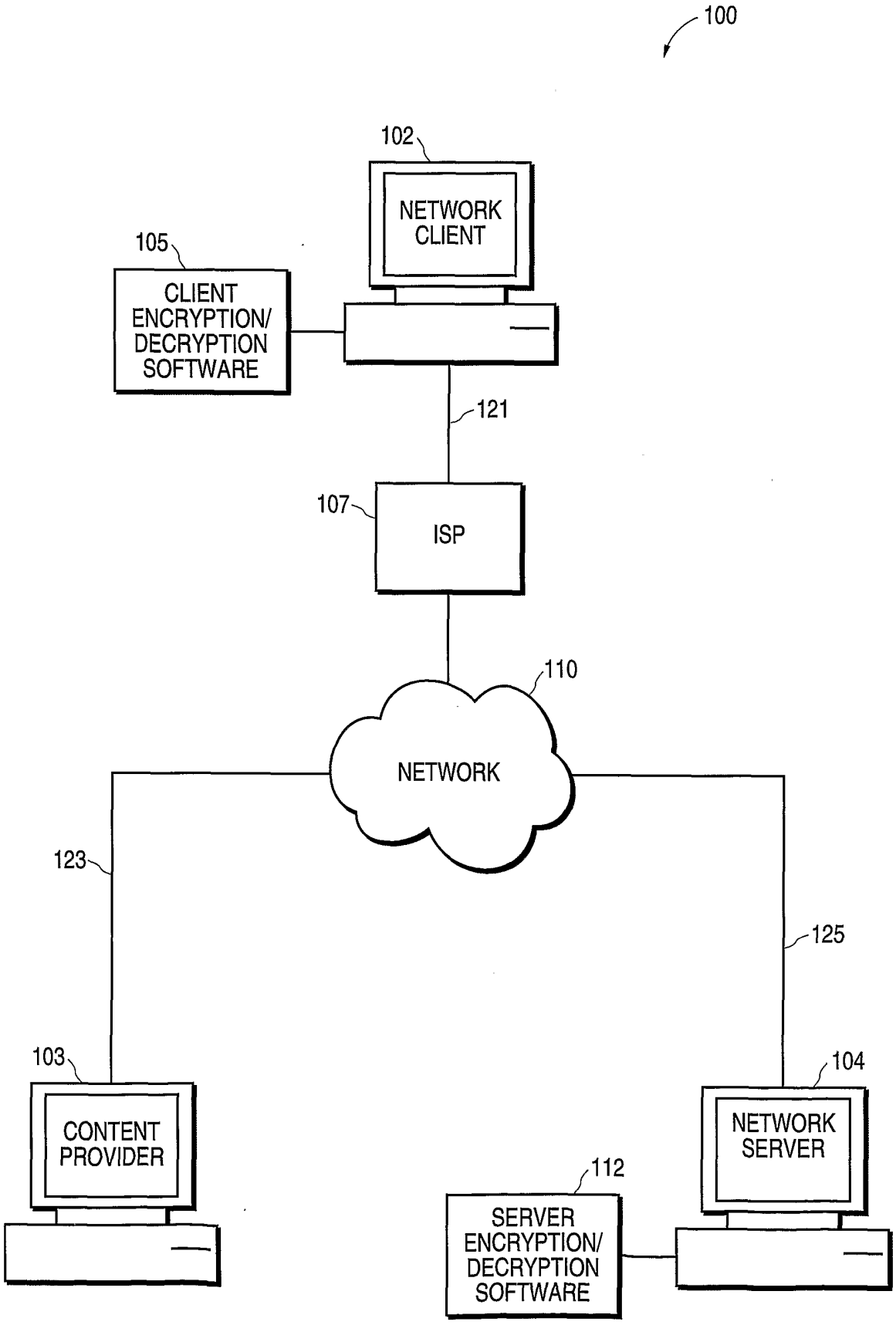

100
102
NETWORK CLIENT
105
CLIENT ENCRYPTION/ DECRYPTION SOFTWARE
121
107
ISP
110
NETWORK
123
125
103
CONTENT PROVIDER
104
NETWORK SERVER
112
SERVER ENCRYPTION/ DECRYPTION SOFTWARE

FIG.1

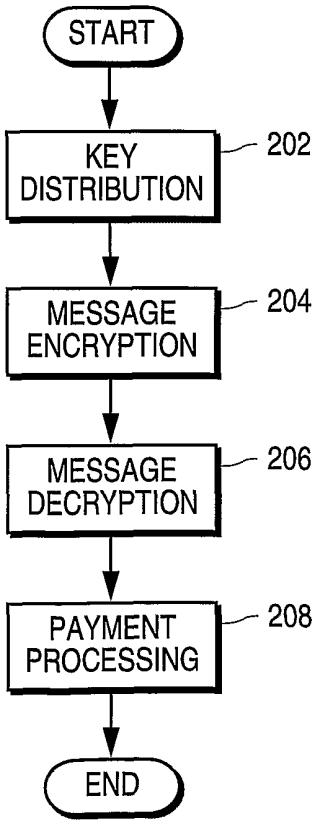

START
KEY DISTRIBUTION
202
MESSAGE ENCRYPTION
204
MESSAGE DECRYPTION
206
PAYMENT PROCESSING
208
END

FIG.2

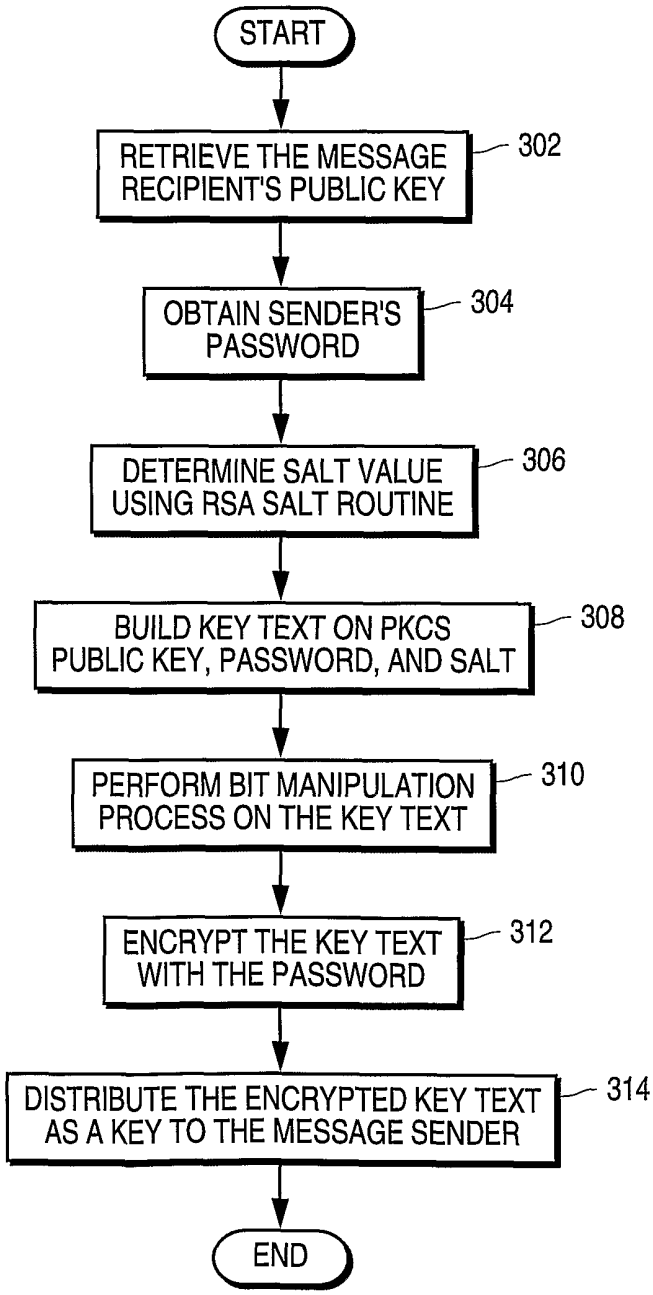

START
RETRIEVE THE MESSAGE RECIPIENT'S PUBLIC KEY
302
OBTAIN SENDER'S PASSWORD
304
DETERMINE SALT VALUE USING RSA SALT ROUTINE
306
BUILD KEY TEXT ON PKCS PUBLIC KEY, PASSWORD, AND SALT
308
PERFORM BIT MANIPULATION PROCESS ON THE KEY TEXT
310
ENCRYPT THE KEY TEXT WITH THE PASSWORD
312
DISTRIBUTE THE ENCRYPTED KEY TEXT AS A KEY TO THE MESSAGE SENDER
314
END

FIG.3

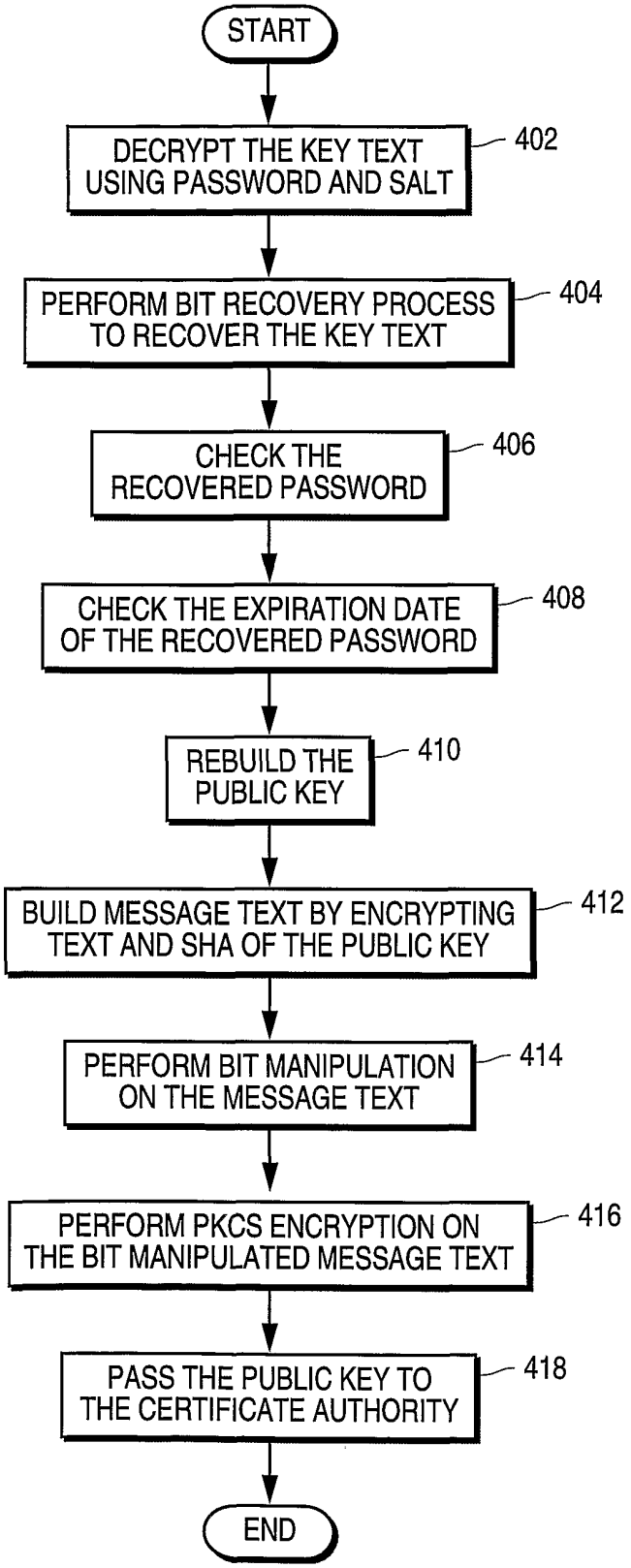

START
DECRYPT THE KEY TEXT USING PASSWORD AND SALT
402
PERFORM BIT RECOVERY PROCESS TO RECOVER THE KEY TEXT
404
CHECK THE RECOVERED PASSWORD
406
CHECK THE EXPIRATION DATE OF THE RECOVERED PASSWORD
408
REBUILD THE PUBLIC KEY
410
BUILD MESSAGE TEXT BY ENCRYPTING TEXT AND SHA OF THE PUBLIC KEY
412
PERFORM BIT MANIPULATION ON THE MESSAGE TEXT
414
PERFORM PKCS ENCRYPTION ON THE BIT MANIPULATED MESSAGE TEXT
416
PASS THE PUBLIC KEY TO THE CERTIFICATE AUTHORITY
418
END

FIG.4

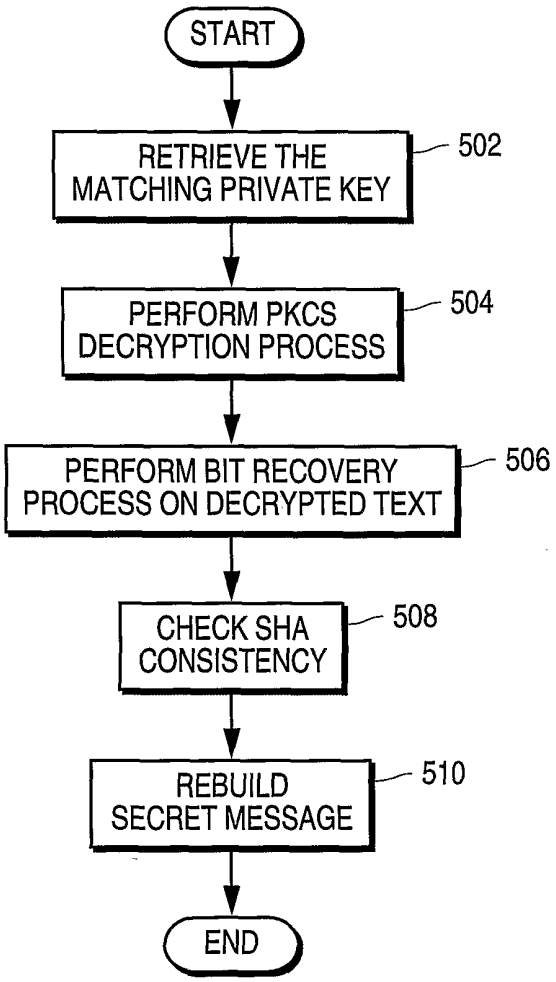

START
RETRIEVE THE MATCHING PRIVATE KEY
502
PERFORM PKCS DECRYPTION PROCESS
504
PERFORM BIT RECOVERY PROCESS ON DECRYPTED TEXT
506
CHECK SHA CONSISTENCY
508
REBUILD SECRET MESSAGE
510
END

FIG.5

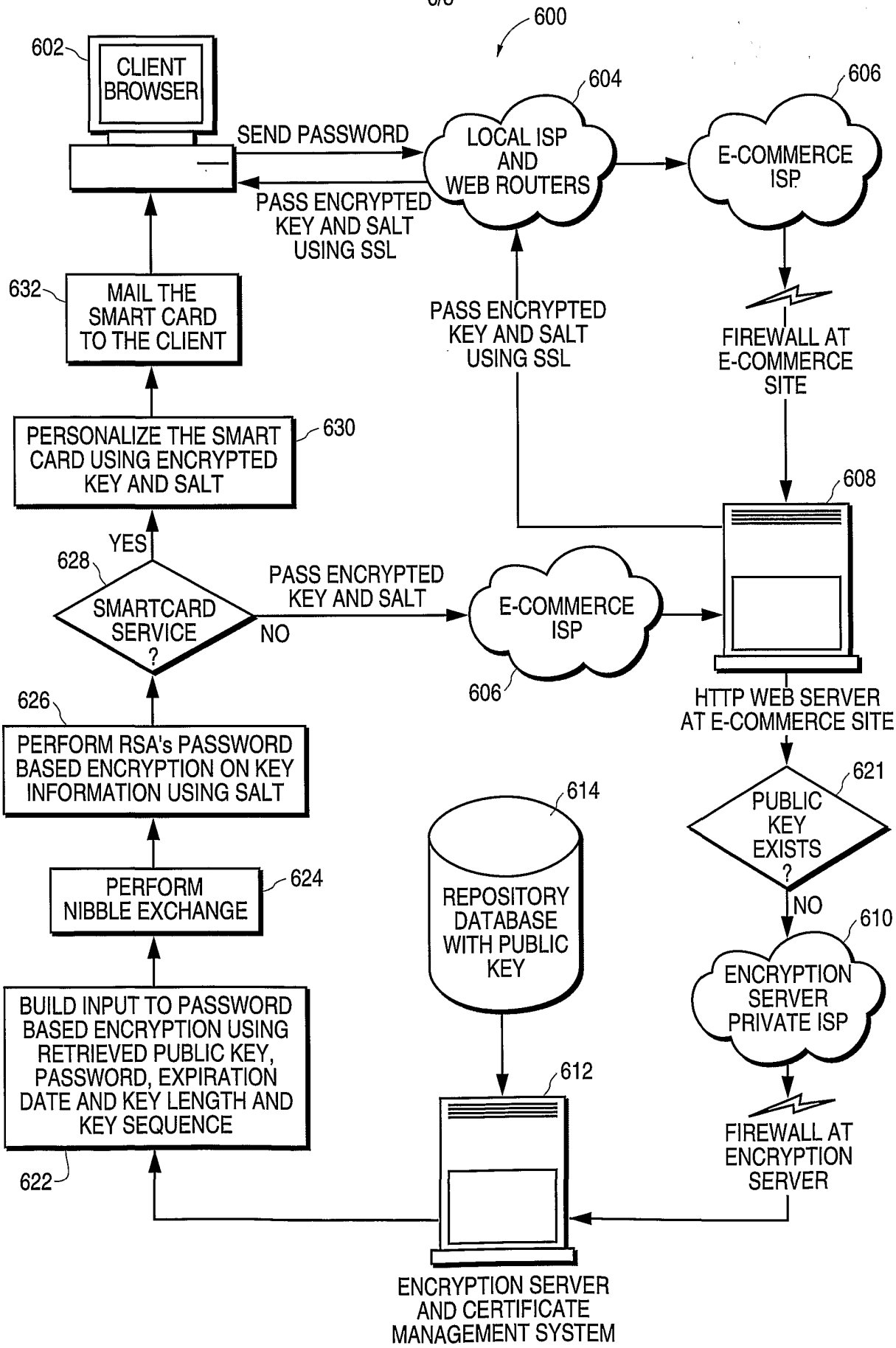

600
602
CLIENT BROWSER
SEND PASSWORD
PASS ENCRYPTED KEY AND SALT USING SSL
604
LOCAL ISP AND WEB ROUTERS
606
E-COMMERCE ISP
632
MAIL THE SMART CARD TO THE CLIENT
PASS ENCRYPTED KEY AND SALT USING SSL
FIREWALL AT E-COMMERCE SITE
630
PERSONALIZE THE SMART CARD USING ENCRYPTED KEY AND SALT
608
YES
628
SMARTCARD SERVICE ?
PASS ENCRYPTED KEY AND SALT
NO
E-COMMERCE ISP
606
HTTP WEB SERVER AT E-COMMERCE SITE
626
PERFORM RSA's PASSWORD BASED ENCRYPTION ON KEY INFORMATION USING SALT
614
REPOSITORY DATABASE WITH PUBLIC KEY
621
PUBLIC KEY EXISTS ?
NO
624
PERFORM NIBBLE EXCHANGE
610
ENCRYPTION SERVER PRIVATE ISP
BUILD INPUT TO PASSWORD BASED ENCRYPTION USING RETRIEVED PUBLIC KEY, PASSWORD, EXPIRATION DATE AND KEY LENGTH AND KEY SEQUENCE
612
FIREWALL AT ENCRYPTION SERVER
622
ENCRYPTION SERVER AND CERTIFICATE MANAGEMENT SYSTEM

FIG.6

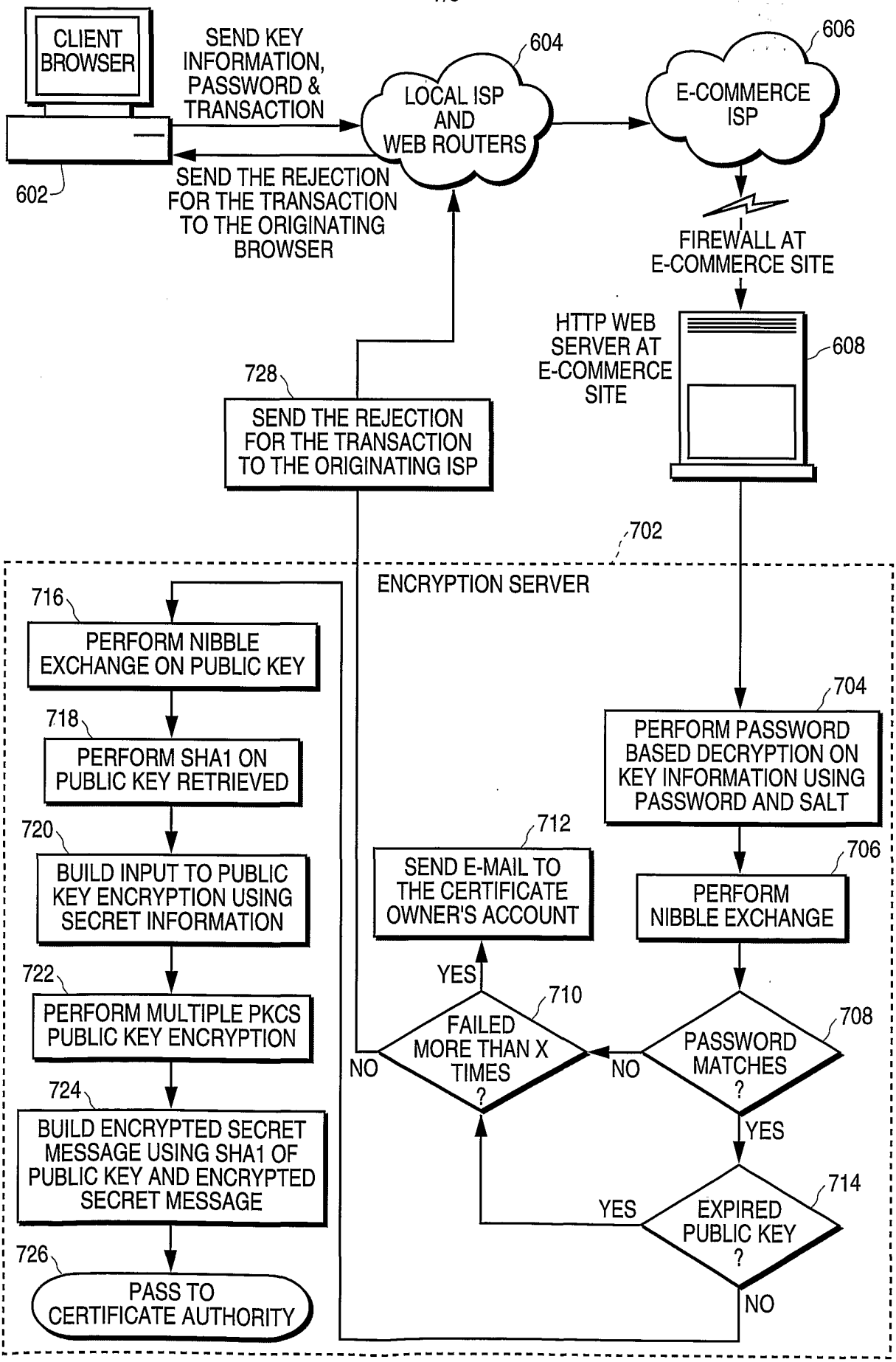

CLIENT BROWSER
602
SEND KEY INFORMATION, PASSWORD & TRANSACTION
SEND THE REJECTION FOR THE TRANSACTION TO THE ORIGINATING BROWSER
604
LOCAL ISP AND WEB ROUTERS
606
E-COMMERCE ISP
FIREWALL AT E-COMMERCE SITE
HTTP WEB SERVER AT E-COMMERCE SITE
608
728
SEND THE REJECTION FOR THE TRANSACTION TO THE ORIGINATING ISP
702
ENCRYPTION SERVER
716
PERFORM NIBBLE EXCHANGE ON PUBLIC KEY
718
PERFORM SHA1 ON PUBLIC KEY RETRIEVED
720
BUILD INPUT TO PUBLIC KEY ENCRYPTION USING SECRET INFORMATION
722
PERFORM MULTIPLE PKCS PUBLIC KEY ENCRYPTION
724
BUILD ENCRYPTED SECRET MESSAGE USING SHA1 OF PUBLIC KEY AND ENCRYPTED SECRET MESSAGE
726
PASS TO CERTIFICATE AUTHORITY
704
PERFORM PASSWORD BASED DECRYPTION ON KEY INFORMATION USING PASSWORD AND SALT
706
PERFORM NIBBLE EXCHANGE
708
PASSWORD MATCHES ?
NO
710
FAILED MORE THAN X TIMES ?
NO
YES
712
SEND E-MAIL TO THE CERTIFICATE OWNER'S ACCOUNT
YES
714
EXPIRED PUBLIC KEY ?
YES
NO

FIG.7

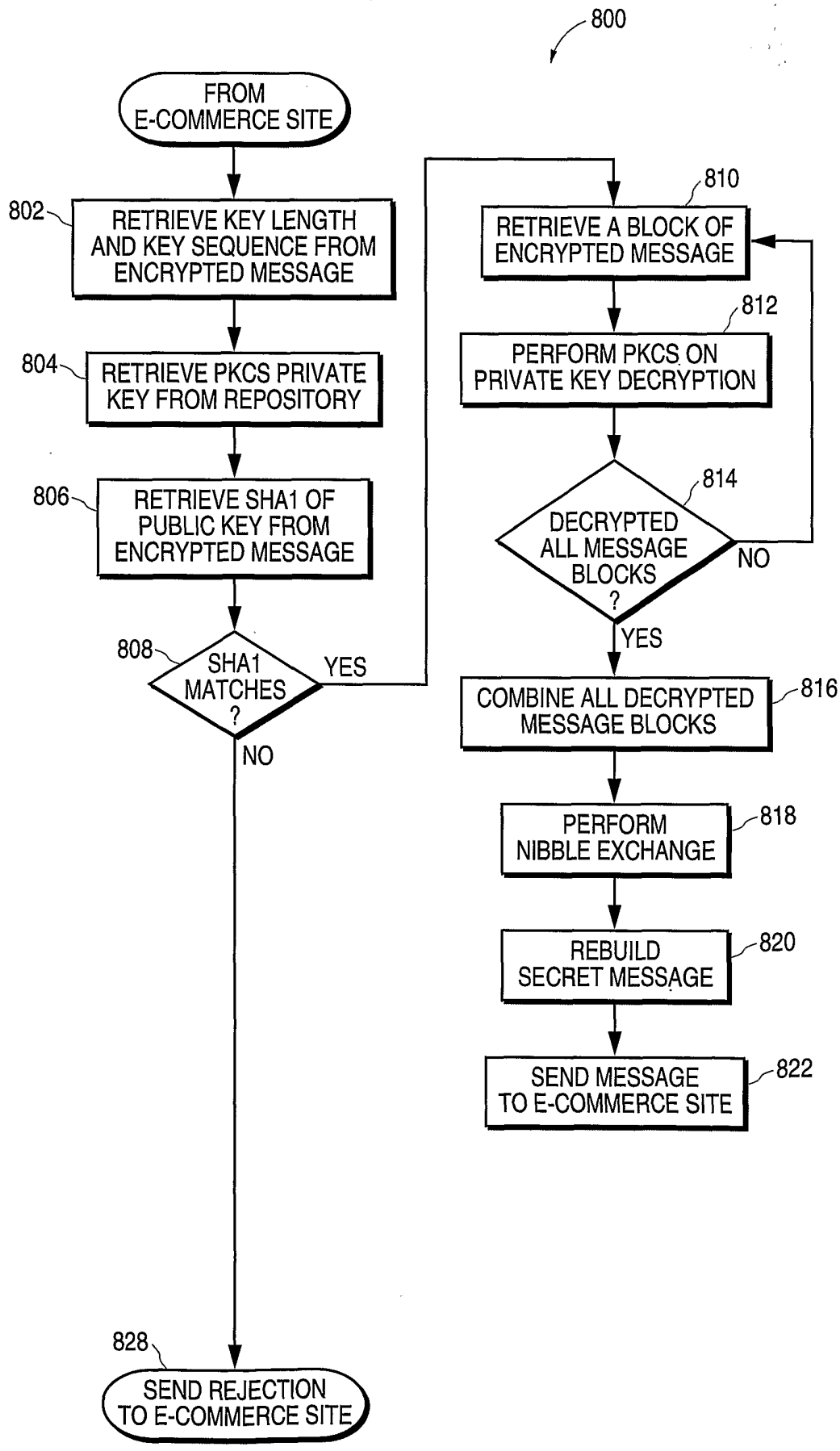

800
FROM E-COMMERCE SITE
802
RETRIEVE KEY LENGTH AND KEY SEQUENCE FROM ENCRYPTED MESSAGE
804
RETRIEVE PKCS PRIVATE KEY FROM REPOSITORY
806
RETRIEVE SHA1 OF PUBLIC KEY FROM ENCRYPTED MESSAGE
808
SHA1 MATCHES ?
YES
NO
810
RETRIEVE A BLOCK OF ENCRYPTED MESSAGE
812
PERFORM PKCS ON PRIVATE KEY DECRYPTION
814
DECRYPTED ALL MESSAGE BLOCKS ?
NO
YES
816
COMBINE ALL DECRYPTED MESSAGE BLOCKS
818
PERFORM NIBBLE EXCHANGE
820
REBUILD SECRET MESSAGE
822
SEND MESSAGE TO E-COMMERCE SITE
828
SEND REJECTION TO E-COMMERCE SITE

FIG.8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/02916

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :HO4L 9/00
US CL : 380/30, 286, 285, 283,; 713/150, 171

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. : 380/30, 286, 285, 283,; 713/150, 171

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,995,624 A (FIELDER et al.) 30 November 1999, col. 3-5, 7 & 22, lines 1-37 1-24.	1-18
Y	US 5,799,086 A (SUDIA) 25 August 1998, col. 4 & 15, lines 1-37 & 1-16.	1-18
A	US 4,596,898 A (PEMMARAJU) 24 June 1986.	1-18
A	US 5,778,069 A (THOMLINSON et al) 07 July 1998.	1-18

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

*	Special categories of cited documents:	"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
"A"	document defining the general state of the art which is not considered to be of particular relevance	"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
"E"	earlier document published on or after the international filing date	"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
"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)	"&"	document member of the same patent family
"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		

Date of the actual completion of the international search
19 MARCH 2001

Date of mailing of the international search report
20 APR 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
TODD M. JACK James R. Matthews
Telephone No. (703) 305-1027

Form PCT/ISA/210 (second sheet) (July 1998)★