Abstract: Embodiments of the present invention provide controlled access to key management servers using store and forward protocols. A computer-implemented method for providing controlled key management includes generating a request indicative of a key management function. The request is received at the first of a number of intermediate parties capable of relaying the request toward a key management server. The key management function is performed subsequent to receiving the request from the last of the intermediate parties which is authorized to provide the request to the key management server. A response to the request is then generated.
BACKGROUND OF THE INVENTION

[0001] The present invention relates to computer systems. More specifically, the present invention relates to techniques for providing controlled access to key management servers.

[0002] In general, a key manager or key management server ("KMS") acts as a secure key vault to store and provide access to one or more keys. A key is a handle on some type of digital asset, that may be encrypted. In general, the key allows a user or computer process to access a digital asset that has been encrypted. Accordingly, it is important to protect and secure keys while allowing access to the keys; the key manager generally provides more services than just encrypting and decrypting a digital asset, including services to label or manage the key stored in a vault.

[0003] FIG. 1 is a simplified diagram of a system 100 for providing key management services in the prior art. In this example, system 100 includes a remote user 110, a communications network 120, a firewall 130, a communications network 140, and a key server 150.

[0004] Typically, remote user 110 requests one or more keys from the key server 150 to obtain access to encrypted objects. Remote user 110 may also perform other operations using key server 150, such as creating new keys or destroying old keys. Key server 150 receives requests from remote user 110, processes the requests, and sends responses to remote user 110.

[0005] In order to access key server 150, remote user 110 typically creates some type of secure tunnel, for example, using TLS, SSL, or another secure means, by which remote user 110 is authenticated to access key server 150. Furthermore, this secure connection is typically a direct connection between remote user 110 and key server 150.

[0006] However, some problems exist when remote user 110 is required to access key server 150 while remote user 110 is physically separated from key server 150, for example, by firewall 130. Firewall 130 is typically configured to provide mitigated access between communications network 120 and communications network 140. In order to establish the direct connection required between remote user 110 and key server 150, one or more holes or
ports must be opened on firewall 130. Firewall 130 then accepts connections from remote user 110 and forwards the connections to key server 150 via communications network 140.

[0007] Additionally, if other external users or devices wish to access key server 150, firewall 130 must be configured to accept these connections and forward these connections from the other external users or devices to key server 150. In some industries and scenarios, these connections may only be required once, or for a very short period of time.

[0008] Thus, to maintain adequate security, an administrator of firewall 130 must be sure to close any holes or ports opened in firewall 130 when the connections are no longer required. This is because for each port opened on firewall 130, a potential security risk is created. If too many ports or holes are opened on firewall 130, the effectiveness and usefulness of the firewall becomes diminished.

[0009] Accordingly, what is desired are improved methods and apparatus for solving the problems discussed above, while reducing the drawbacks discussed above.

BRIEF SUMMARY OF THE INVENTION

[0010] The present invention relates to techniques for providing controlled access to key management servers. In short, embodiments of the present invention provide controlled access to key management servers using store and forward protocols. By designating one or more hosts or intermediate parties configured and authorized to accept connections or requests from clients addressed to a key management server, the one or more hosts or intermediate parties can deliver requests indicative of key management functions to a protected key management server. Accordingly, rather than opening access to the key management server for each device, network, etc., that needs access to key management services, communication may be performed through a number of intermediate parties (i.e., one or more) that act as gatekeepers.

[0011] Thus, a number of intermediate parties, or even only the last of a number of intermediate parties, are required to have privileges to access a protected key management server. Accordingly, embodiments of the present invention may employ existing communications services to provide mitigated access to key management servers, such as e-mail or Internet messaging, that may already include mechanisms for security and authentication and that further provide scalability for large numbers of users.
[0012] In various embodiments, a computer-implemented method for providing controlled key management includes generating a request indicative of a key management function. The request is received at the first of a number of intermediate parties capable of relaying the request toward a key management server. The key management function is performed subsequent to receiving the request from the last of the intermediate parties that is authorized to provide the request to the key management server. A response to the request is then generated.

[0013] In some embodiments, receiving the request at the first of a number of intermediate parties capable of relaying the request to a key management server includes receiving the request at the first of the number of intermediate parties using a store and forward protocol. Receiving the request at the first of a number of intermediate parties capable of relaying the request toward a key management server may include receiving the request at an e-mail server. The number of intermediate parties may be one, such that receiving the request at the first of a number of intermediate parties capable of relaying the request toward a key management server may include receiving the request at the last of the number of intermediate parties.

[0014] In various embodiments, the response to the request is received at the first of a number of intermediate parties capable of relaying the response to a client. The last of the number of intermediate parties may be configured to deliver the response to the client. The key management function may include at least one of a create operation, a store operation, a retrieve operation, a find operation, a disable operation, a destroy operation, and a modify operation.

[0015] In some embodiments, the request may be encrypted. The request may also be digitally signed. The request may then be authenticated in response to the digital signature and may be decrypted based on a positive determination that the request is authentic. The response may also be encrypted and digitally signed. The response may be authenticated in response to the digital signature and decrypted based on a positive determination that the request is authentic.

[0016] In some embodiments, a computer program product is stored on a computer-readable medium for providing controlled key management. The computer program product includes code for generating a request indicative of a key management function, code for receiving the request at the first of a number of intermediate parties capable of relaying the
request toward a key management server, code for performing the key management function subsequent to receiving the request from the last of the intermediate parties that is authorized to provide the request to the key management server, and code for generating a response to the request.

5 [0017] In various embodiments, a system for providing controlled key management includes a number of intermediate parties and a key management server. The first of the number of intermediate parties may be configured to receive a request indicative of a key management function from a client. The last of the number of intermediate parties may be authorized to provide the request to one or more key management servers. The key management server may be configured to receive the request from the last of the number of intermediate parties, perform the key management function, and generate a response to the request.

10 [0018] In still further embodiments, a system for secured key management includes a key management server and a first server. The first server is communicatively positioned between the key management server and a client. The first server is configured to receive a request addressed to the key management server from the client, the request indicative of a key management function, deliver the request to the key management server if the first server is authorized to deliver a request from a client to the key management server, and relay the request to a second server communicatively positioned between the key management server and the client if the first server is not configured to access the key management server.

15 [0019] The first server may receive a response to the request addressed to the client, deliver the response to the client if the first server is configured to access the client, and relay the response to a third server if the first server is not configured to access the client.

20 [0020] In various embodiments, a system for providing controlled key management includes a processor and memory. The memory is coupled to the processor and configured to store a plurality of code modules that when executed by the processor cause the processor to receive a request addressed to a key management server from a client, the request indicative of a key management function, deliver the request to the key management server if permitted to deliver requests from clients to the key management server, and relay the request to a first host communicatively positioned between the key management server and the client if not permitted to access the key management server.
The processor may receive a response to the request addressed to the client, deliver
the response to the client if capable of accessing the client, and relay the response to a second
host communicatively positioned between the key management server and the client if not
capable of accessing the client.

A further understanding of the nature and the advantages of the inventions disclosed
herein may be realized by reference of the remaining portions of the specification and the
attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to more fully understand the present invention, reference is made to the
accompanying drawings. Understanding that these drawings are not to be considered
limitations in the scope of the invention, the presently described embodiments and the
presently understood best mode of the invention are described with additional detail through
use of the accompanying drawings.

FIG. 1 is a simplified diagram of a system for providing key management
services in the prior art;

FIG. 2 is a block diagram of a system for providing controlled access to a key
management server in some embodiments according to the present invention;

FIGS. 3A and 3B are a simplified flowchart of a method for providing controlled
access to key management server in some embodiments according to the present invention;

FIG. 4 is a block diagram of a system for providing controlled access to a key
management server in some embodiments according to the present invention;

FIG. 5 is a message sequence chart illustrating key management access in the
system of FIG. 4 in some embodiments according to the present invention;

FIG. 6 is a block diagram of a system for providing controlled access to a key
management server in some embodiments according to the present invention;

FIG. 7A, 7B, and 7C are a message sequence chart illustrating key management
access in the system of FIG. 6 in some embodiments according to the present invention; and

FIG. 8 is a simplified block diagram of a computer system that may incorporate
embodiments of the present invention.
DETAILED DESCRIPTION OF THE INVENTION

[0032] The present invention relates to techniques for providing controlled access to key management servers. In various embodiments, controlled access to a key management server is provided using a store and forward protocol. Some examples of store and forward protocols are file transfer protocol (FTP), simple mail transfer protocol (SMTP), instant or Internet messaging (IM), and the like. A key management server may be physically or electronically secured while allowing requests for keys and other key management functions to be delivered to the key management server using a number (one or more) of intermediate parties.

[0033] The intermediate parties use store and forward protocols to deliver the request to the key management server. Accordingly, a firewall or other network access control mechanism may be configured to allow connections only from those intermediate parties authorized to deliver the request to the key management server.

[0034] Thus, a number of intermediate parties, or even only the last of a number of intermediate parties, are required to have privileges to access a key management server. Accordingly, embodiments of the present invention may employ existing communications services, such as e-mail or Internet messaging, that already include mechanisms for security and authentication and that also provide scalability for large numbers of users, to provide mitigated access to key management servers.

[0035] Additionally, the key management server may deliver responses to client requests also using store and forward protocols. This allows the functionality of the key management server to scale and provide services to additional unanticipated clients or other devices, while still providing security and control over access to the key management server.

[0036] FIG. 2 is a block diagram of a system 200 for providing controlled access to a KMS 240 in some embodiments according to the present invention. In this example, system 200 includes a private network 210, a firewall 220, a public network 230, KMS 240, one or more local communications servers 250, one or more local users 260, one or more remote communications servers 270, and one or more remote users 280.

[0037] In this example, private network 210 is link via firewall 220 to public communications network 230. Private network 210 is linked to KMS 240, the one or more
local communications servers 250, and the one or more local users 260. Public network 230 is linked to remote communications servers 270 and remote users 280.

[0038] Private network 210 may be any communications network or link for exchanging data. Some examples of private network 210 include local area networks (LANs), wide area networks (WANs), metropolitan area networks (MANs), wireless local area networks (WLANs), and the like. Private network 210 may include firewalls, network address translation (NAT) devices, network intrusion detection devices, and the like, to manage access and security.

[0039] Firewall 220 includes hardware and/or software elements that permit, deny or proxy data connections between private network 210 and public network 230. In general, firewall 220 controls traffic between computer networks with different zones of trust. Typically, public network 230 (e.g., the Internet) is a zone with little or no trust, and private network 210 (e.g., an internal network) is a zone with high trust. Accordingly, firewall 220 attempts to control interfaces between zones of differing trust levels through the enforcement of a security policy and connectivity model.

[0040] Public network 230 is any communications network or link for exchanging data. Some examples of public network 230 include LANs, WANs, MANs, WLANs, hotspots, public leased lines, public networks, the Internet, and the like.

[0041] KMS 240 includes hardware and/or software elements that provide one or more key management functions. Some examples of key management functions are key generation, key destruction, key retrieval, key validation, key labeling, and the like. In general, a key is a handle for accessing a digital object or asset. The digital object may be encrypted or otherwise have access encumbered by a security scheme. One example of KMS 240 is "CryptoStor KeyVault" provided by NeoScale Systems, Inc. of Milpitas, California.

[0042] Local communications servers 250 include hardware and/or software elements that provide communications services. Some examples of communications services are e-mail, instant or Internet messaging, file transfer, file storage, short message service (SMS), and the like. Local communications servers 250 may use various protocols, such as store and forward protocols, to communicate between other devices. In general, a store and forward protocol is any communication protocol in which a direct connection is not required for communication between one or more hosts. Typically, one or more intermediate parties...
between a first host and a second host relay communications and/or exchange data between
the two hosts.

[0043] Local users 260 include hardware and software elements that provide general and
specific purpose computing to a user. Some examples include personal computers (PCs),
desktops, laptops, handheld devices, mainframes, microcomputers, workstations, thin-clients,
and the like.

[0044] Remote communications servers 270 include hardware and/or software elements
that provide communications services via public network 230. Remote users 280 include
hardware and software elements that provide general and specific purpose computing to a
user.

[0045] In general, firewall 220 provides mitigated access to KMS 240, local
communications servers 250, and local users 260 for connections from public network 230.
For example, firewall 220 may accept, reject, drop, deny, and proxy connections from public
network 230 to services behind the firewall that are attached to private network 210. Firewall
220 may also perform Network Address Translation (NAT), such as network masquerading,
native address translation, or IP-masquerading.

[0046] In one example of operation, a local user 260 generates a request indicative of a key
management function to obtain access to an encrypted digital asset. Local user 260 then
forwards the request via private network 210 to KMS 240. KMS 240 authenticates the
request, and generates a response based on performance of the key management function
indicated in the request from local user 260. KMS 240 then forwards a response to local user
260 that may include a key to allow the client to access the encrypted digital asset.

[0047] In another example of operation, a local user 260 generates a request indicative of a key
management function to obtain access to an encrypted digital asset. Local user 260 then
forwards the request via private network 210 to one of local communications servers 250.
The one particular local communications server 250 may forward the request to KMS 240, or
forward the request to another local communications server 250. Eventually, one of the local
communications servers 250 may be authorized to provide requests to KMS 240. KMS 240
then authenticates the request, and generates a response based on performance of the key
management function indicated in the request from the local user 260.
Typically, if one or more remote users desire to access KMS, ports are opened on firewall, and forwarded to KMS. In one example of operation, system employs one or more intermediate parties (e.g., local and remote communications servers) to convey requests indicative of key management functions to KMS without unnecessarily opening more ports on a firewall. Accordingly, the one or more intermediate parties can store and forward requests indicative of key management functions to the KMS while still maintaining the security of private network. Additionally, local and remote communications servers may be configured to deliver responses from the KMS to local users and remote users.

For example, a remote user generates a request indicative of a key management function to obtain access to an encrypted digital asset. Remote user then forwards the request via public network to one of remote communications servers. The particular remote communications server may forward the request to another remote communications server. Eventually, one of remote communications server is authorized to provide requests to firewall. At firewall, one or more particular local communications servers may be configured to receive requests from remote communications servers.

As discussed above, the one or more particular local communications servers may forward the request to KMS, or forward the request to another local communications server. Eventually, one of local communications server is authorized to provide requests to KMS. KMS then authenticates the request, and generates a response based on performance of the key management function indicated in the request from the remote user.

In some embodiments, KMS responds directly to local users or remote users. In various embodiments, KMS responds to local users or remote users using a number of intermediate parties (e.g., local communications servers and remote communications servers). KMS may respond using the same store and forward protocol in which a request was received. KMS may also respond using a different protocol that that used to transmit the request.

In various embodiments, system provides a drop box using local communications servers and remote communications servers in which local users or remote users can generate a request, which may be digitally signed and encrypted,
and placed the request in the drop box via one or more intermediate parties. At a plurality of times, system 200 checks the drop box to determine whether any requests have been deposited, retrieves any deposited requests, and forwards the any deposited requests to KMS 240. The drop box may be provided by an FTP server in which the drop box is right only and cannot be read from by unauthorized devices. The drop box may also be provided by secure file copy protocols (SCP).

In various embodiments, system 200 enables existing services that provide store and forward protocol capabilities, such as e-mail to be leveraged in combination with the KMS 240. Typically, e-mail servers include user, host, and network authentication mechanisms, spyware and virus filtering, and already act as gateways into a corporate or private network (e.g., private network 210). Accordingly, KMS 240 may be provided an e-mail address to which requests may be forwarded. E-mail communications received that are addressed to the e-mail address associated with KMS 240 may be logged, filtered, authenticated, virus scanned and spam-checked, further encrypted or tunneled, and the like, before reaching KMS 240.

Thus, system 200 provides controlled access to KMS 240 using store and forward protocols. System 200 enables users both local and remote to have access to key management functions without compromising security of private network 210. System 200 further provides mechanisms that allow unanticipated or additional devices to be added anytime to access KMS 240 in a secure manner.

FIGS. 3A and 3B are a simplified flowchart of a method for providing controlled access to KMS 240 in some embodiments according to the present invention. The processing depicted in FIGS. 3A and 3B may be performed by software modules (e.g., instructions or code) executed by a processor of a computer system, by hardware modules of the computer system, or by combinations thereof. FIG. 3A begins in step 300.

In step 305, a request is generated indicative of a key management function. A request is any signal, message, instruction, and the like. A key management function is an act, operation, or procedure performed by a key manager. Some examples of key management functions are create operations, delete operations, destroy operations, label operations, retrieve operations, update operations, and the like. In one example, remote user 280 generates a request addressed to KMS 240 to request a key to access an encrypted digital object.
In step 310, the request is received at the first of a number of intermediate parties capable of relaying requests toward a key management server. In general, an intermediate party is any device, host, process, and the like, that receives information from a first host and forwards the information to a second host. In one example, remote user 280 delivers the request to one or more of remote communications servers 270.

In step 315, a determination is made whether an intermediate party (e.g., the first of the number of intermediate parties) is authorized to provide the request to the key management server. Some examples of mechanisms by which the determination may be made include determinations based upon IP address or other network addresses, subnet addresses, Internet service providers, port ranges, MAC or hardware addresses, date and time limitations, and the like.

Based on a negative determination, in step 320, the request is received at the next intermediate party capable of relaying the request toward the key management server. For example, a given remote communications server 270 may forward the request to another remote communications server 270, or to one of local communications servers 250.

Based on a positive determination, in step 325, the request is received at the key management server (e.g., KMS 240) from the last of the intermediate parties. In some embodiments, the last of the number of intermediate parties may also be the first of the number of intermediate parties to which the request was delivered.

FIG. 3A ends in step 330, where processing continues in FIG. 3B. FIG. 3B begins in step 335.

In step 340, the key management function indicated in the request is performed. In step 345, a response is generated indicative of performance of the key management function. In other words, KMS 240 may generate a response including the results of the key management function, e.g., including the key requested by the remote user 280 to obtain access to the encrypted digital object. In various embodiments, the response may be encrypted and digitally signed by KMS 240.

In step 350, the response is received at the first of a number of intermediate parties capable of delivering the response to the client. For example, KMS 240 may deliver the response to an e-mail server, an instant or Internet messaging server, an FTP server, and the like. In some embodiments, the response may be delivered directly to the client. In step 355,
a determination is made whether an intermediate party (e.g., the first of the number of intermediate parties) is capable of delivering the response to the client.

[0064] Based on a negative determination, in step 360, the response is received at the next intermediate party capably of relaying the response toward the client. For example, a given local communications server 250 may forward the request to another local communications server 250, or to one of remote communications servers 270.

[0065] Based on a positive determination, in step 365, the response is received at the client (e.g., remote user 280) from the last of the intermediate parties. In some embodiments, the last of the number of intermediate parties may also be the first of the number of intermediate parties to which the request was delivered. FIG. 3B ends in step 370.

[0066] Accordingly, system 200 provides controlled access to KMS 240 using store and forward protocols. By designating one or more hosts (e.g., local and remote communication servers 250 and 270) that are configured and authorized to accept connections from clients for a key management server, the one or more hosts can be configured to deliver requests indicative of key management functions to the key management server behind the firewall. Accordingly, rather than opening access to the key management server for each device needing access to key management services, devices can communicate through one or more intermediate parties that act as gatekeepers.

[0067] FIG. 4 is a block diagram of a system 400 for providing controlled access to a KMS 460 in some embodiments according to the present invention. In this example, system 400 includes a client 410, a communications network 420, a firewall 430, a communications network 440, an intermediate party 450, and KMS 460.

[0068] Communications network 420 is linked to client 410 and firewall 430. Communications network 440 is linked to firewall 430, intermediate party 450, and KMS 460.

[0069] In general, firewall 430 provides mitigated access to communications network 440 from communications network 420. For example, firewall 430 intercepts connections for FTP communication (e.g., on control port 21), and forwards the connections via communications network 440 to intermediate party 450. In another example, firewall 430 intercepts connections related to Internet relay chat (IRC), instant or Internet messaging, or
other messaging services, and forwards the connections via communications network 440 to intermediate party 450.

[0070] Intermediate party 450 includes hardware and/or software components for providing communication services. For example, intermediate party 450 may provide simple anonymous file transfers, authenticated file transfers, and/or secure encrypted file transfers. Intermediate party 450 may further include user authentication mechanisms and user access policies. While only one intermediate party 450 is shown, any number of intermediate parties may be included in system 400.

[0071] KMS 460 includes hardware and/or software elements that provide key management functions. One example of KMS 460 is "CryptoStor KeyVault" provided by NeoScale Systems, Inc. of Milpitas, California.

[0072] In one example of operation, clients (e.g., client 410) connected to communications network 420 generate and transmit key requests to KMS 460 via communications network 420. Firewall 430 intercepts connections from communications network 420, and accepts, proxies, forwards, or rejects the connections. Connections containing key requests addressed to KMS 460, such as on a particular port or port range, or at a particular e-mail address or other network identifier, are forwarded to intermediate party 450. Intermediate party 450 delivers key requests to KMS 460. For example, intermediate party 450 may push the requests to KMS 460, or simply wait to be polled by KMS 460 to have key requests pulled by KMS 460. KMS 460 processes key requests and returns the results to the requester. One example of operation of system 400 is described further with respect to FIG. 5.

[0073] In various embodiments, key requests addressed to KMS 460 may need to go through one or more intermediate parties 450. An intermediate party that is not authorized to deliver a key request or cannot deliver key requests directly to KMS 460 may instead forward key requests to another intermediate party 450 who is authorized to or can directly forward key requests to KMS 460, or who can forward key requests to yet another intermediate party. For example, in the case of communicating via e-mail, key requests from a user may have to pass through a corporate gateway, then to an internet service provider (ISP) for that company, then to another ISP (of the network associated with KMS 460), then to an e-mail gateway for the network associated with the KMS 460, and finally to KMS 460 itself.

[0074] FIG. 5 is a message sequence chart 500 illustrating key management in system 400 in some embodiments according to the present invention. FIG. 5 begins in step 505.
In step 505, client 410 generates a request for a key to decrypt an object. For example, a user of client 410 may be attempting to access an encrypted data file. In some embodiments, client 410 encrypts the request, and digitally signs the encrypted request.

In step 510, client 410 sends the request addressed to KMS 460 over communications network 420 to intermediate party 450. In this example, firewall 430 intercepts connections from communications network 420, and determines whether to accept the connections and forward them to intermediate party 450, or reject the connections. If the connections are accepted, firewall 430 forwards the connections, and thereby the request sent by client 410, to intermediate party 450.

In step 515, intermediate party 450 processes the request to determine whether to deliver the request. For example, intermediate party 450 may require that the user of client 410 enter a username and password to deliver the request to intermediate party 450. In some embodiments, intermediate party 450 requires that client 410 be authenticated using a digital signature. Other authentication mechanisms and user security policies may be employed by intermediate party 450. Intermediate party 450 may also scan the request for viruses, spyware, spam, and the like.

In step 520, if intermediate party 450 is not authorized to deliver the request to KMS 460, intermediate party 450 may reject the request step 525. For example, intermediate party 450 may accept anonymous incoming connections for other services, e.g., public file transfers, however anonymous requests placed into directories designated for key management may be rejected. Intermediate party 450 may also accept e-mail designated for KMS 460, however the request must be included in an encrypted e-mail attachment. E-mails not including the encrypted e-mail attachment are dropped or rejected by intermediate party 450.

Intermediate party 450 may also forward the request to another host or device capable of relying the request toward a key management server. In various embodiments, intermediate party 450 may determine that it cannot deliver directly to KMS 460. Thus, the request might go through a chain of intermediate parties before delivering to one who is authorized to deliver the request to KMS 460.

If intermediate party 450 is authorized to deliver the request to KMS 460, in step 530, intermediate party 450 delivers the request to KMS 460. For example, a script periodically executed on intermediate party 450 may forward uploaded requests to KMS 460.
Alternatively, KMS 460 may periodically poll intermediate party 450 for uploaded requests. Any variety of push and pull mechanisms may be employed to deliver the request to the KMS 460.

[0081] In step 535, KMS 460 processes the request to generate a response including the key. KMS 460 may perform one or more key management functions to retrieve, create, or obtain the key requested by client 410. In step 540, KMS 460 transmits the response addressed to client 410 via communications network 440.

[0082] In some embodiments, KMS 460 may encrypt and digitally sign the response that includes the key. KMS 460 may use encryption programs such as PGP, GPG, and the like, to encrypt the response. KMS 460 may also use digital signature mechanisms, or electronic signatures, to digitally sign the response.

[0083] In step 545, client 410 receives and processes the response to obtain the key. For example, client 410 may verify the digital signature of KMS 460 and decrypt the response to obtain the key. In step 550, client 410 decrypts the object using the key obtained in step 545 to gain access to the object. Message sequence chart 500 ends in FIG. 5 at step 550.

[0084] Accordingly, firewall 430 in system 400 does not have to provide direct access to KMS 460, while intermediate party 450 may be considered an intermediate party authorized to deliver requests indicative of key management functions to key management server. By mitigating access, KMS 460 may be more securely isolated from attack, while still providing key management to local and remote users.

[0085] However, in this example KMS 460 is configured to respond directly to client 410. In various embodiments, KMS 460 may be configured to also use one or more intermediate parties to indirectly forward responses to requests indicative of key management functions. Thus, in various embodiments, KMS 460 is configured to deliver responses to intermediate party 450, which further delivers responses to client 410. In some embodiments, one or more intermediate parties may be used to deliver responses to users. It may be possible that responses may take a different route returning to a user than the route used to deliver key requests to the KMS 460.

[0086] FIG. 6 is a block diagram of a system 600 for providing controlled access to a KMS 650 in some embodiments according to the present invention. System 600 includes a client 610, a remote e-mail server 620, a firewall 630, a local e-mail server 640, and KMS 650. For
the sake of simplicity, intermediate public and private communications networks have not been shown.

[0087] In this example, remote e-mail server 620 is linked to client 610 and to firewall 630. Local e-mail server 640 is linked to firewall 630 and to KMS 650.

[0088] In general, firewall 630 provides mitigated access to local e-mail server 640. For example, firewall 630 intercepts connections related to e-mail communications, and drops or denies all other types of connections. Firewall 630 then forwards connections related to e-mail communications to local e-mail server 640. In this example, connections from remote e-mail server 620 may be forwarded through firewall 630 to local e-mail server 640.

[0089] Remote e-mail server 620 and local e-mail server 640 include hardware and or software elements that provide e-mail communications services, such as Microsoft Exchange, IBM Lotus Notes, Linux and Unix messaging systems (e.g., postfix and sendmail), and the like. Remote e-mail server 620 and local e-mail server 620 may be configured to use a variety of protocols such as SMTP, POP3, IMAP, UUCP, rsync, and the like.

[0090] KMS 650 is configured to receive requests using local e-mail server 640, and to forward responses again using local e-mail server 640. KMS 650 may be provided a mailbox, in which mail sent to an e-mail address assigned to the KMS 650 is deposited into the mailbox associated with KMS 650. One example of operation of system 600 is described further with respect to FIGS. 7A, 7B, and 7C.

[0091] FIG. 7A, 7B, and 7C are a message sequence chart 700 illustrating key management in system 600 in some embodiments according to the present invention. Message sequence chart 700 begins in FIG. 7A in step 702.

[0092] In step 702, client 610 generates a request indicative of a key management function. In step 704, client 610 encrypts and digitally signs the request. In step 706, client 610 sends the request to remote e-mail server 620 addressed to KMS 650.

[0093] In some embodiments, a user of client 610 may attach an encrypted and digitally signed key request to an e-mail. The user then may transmit the e-mail addressed to KMS 650 using a mail client, such as Outlook. In other embodiments, a computer process executing on client 610 using SMTP communicates with e-mail server 620 to deliver an encrypted and digitally signed request that has been automatically generated for the user of client 610 in response to a local operation to access a protected object.
In step 708, remote e-mail server 620 determines whether it is authorized to deliver the request to KMS 650. For example, remote e-mail server 620 may determine whether it hosts the domain associated with the e-mail address of KMS 650. Based on a positive determination that remote e-mail server 620 is authorized, remote e-mail server 620 would send the request directly to KMS 650 (not shown). However, in this example, remote e-mail server 620 is not authorized to deliver the request to KMS 650 because, in this example, remote e-mail server 620 neither hosts the domain associated with the e-mail address of KMS 650 nor has a direct connection to KMS 650.

In some examples, in step 710, if remote e-mail server 620 is authorized to deliver the request, in step 712, remote e-mail server 620 delivers the request to KMS 650. FIG. 7A then ends processing in step 714, where message sequence chart 700 continues in step 736 of FIG. 7C.

However, in this example, in step 716, remote e-mail server 620 determines one or more e-mail exchanges related to the e-mail address associated with KMS 650. The one or more e-mail exchanges may indicate that local e-mail server 640 is configured to receive e-mail for the KMS 650. In this example, local e-mail server 640 is configured to be the destination indicated by the one or more e-mail exchanges. In step 718, remote e-mail server 620 forwards the encrypted and digitally signed request to local e-mail server 640. FIG. 7A ends in step 720, where message sequence chart 700 continues in step 722 of FIG. 7B.

Referring to FIG. 7B, in this example, firewall 630 intercepts e-mail communications (e.g., from remote e-mail server 620), and forwards the e-mail communications to local e-mail server 640. In step 724, local e-mail server 640 determines whether it is authorized to deliver the request to KMS 650. For example, local e-mail server 640 may determine whether it recognizes the e-mail address used to deliver the request.

Local e-mail server 640 may perform other policy-based, host-based, user-based, and network checks to determine whether it is authorized to deliver requests to KMS 650. Local e-mail server 640 may also perform virus scanning, spam filtering, and other checks on the request to determine the validity, authenticity, and the like, of the request.

In step 726, if local e-mail server 640 is not authorized to deliver the information to KMS 650, in step 728, local e-mail server 640 discards the request. Local e-mail server 640 may also reject the request, or accept the request and subsequently discard the request without generating an error message.
In step 726, if local e-mail server 640 is authorized to deliver the request to KMS 650, in step 730, local e-mail server 640 stores the request in a mailbox associated with KMS 650. In step 732, at one of a plurality of times, KMS 650 authenticates itself to local e-mail server 640 and retrieves requests stored in the mailbox associated with KMS 650. FIG. 7B ends in step 734, where message sequence chart 700 continues in step 736 of FIG. 7C.

Referring now to FIG. 7C, in step 738, KMS 650 processes the digital signature associated with the request to authenticate the information as received from client 610. In step 740, if the digital signature is not valid, in step 742, KMS 650 ignores, rejects, or otherwise discards the request.

In step 740, if the digital signature is valid, in step 744, KMS 650 decrypts the request. In step 746, KMS 650 performs one or more key management functions indicated by the request. For example, key management server may generate a key, destroy a key, obtain a key, update a key, update metadata associated with the key, generate a label for a key, and the like.

In step 748, KMS 650 generates a response indicative of performance of the one or more key management functions. In step 750, KMS 650 encrypts and digitally signs the response indicating the results of the key management functions. In step 752, KMS 650 delivers the encrypted and digitally signed response indicating the results of the key management functions to local e-mail server 640 addressed to client 610. For example, KMS 650 may address the response to an e-mail address associated with a user of client 610.

In step 754, local e-mail server 640 forwards the response to remote e-mail server 620 for delivery to client 610. In step 756, remote e-mail server 620 stores the response in a mailbox for client 610. For example, remote e-mail server 620 may interface with an Exchange, POP3, IMAP, or Web-mail server.

In step 758, client 610 retrieves the response delivered to remote e-mail server 620. For example, the user of client 610 may manually retrieve the response from remote e-mail server 620. In another example, a computer process executing on client 610 may periodically poll remote e-mail server 620 to determine whether the response has arrived. In step 760, client 610 processes the response to obtain the results of the one or more key management functions. For example, client 610 may obtain a key to decrypt a digital object. Message sequence chart 700 ends in FIG. 7C at step 760.
FIG. 8 is a simplified block diagram of a computer system 800 that may be used to practice embodiments of the present invention. As shown in FIG. 8, computer system 800 includes a processor 805 that communicates with a number of peripheral devices via a bus subsystem 810. These peripheral devices may include a storage subsystem 815, comprising a memory subsystem 835 and a file storage subsystem 855, user interface input devices 820, user interface output devices 825, and a network interface subsystem 830.

Bus subsystem 810 provides a mechanism for letting the various components and subsystems of computer system 800 communicate with each other as intended. Although bus subsystem 810 is shown schematically as a single bus, alternative embodiments of the bus subsystem may utilize multiple buses.

Network interface subsystem 830 provides an interface to other computer systems, and networks, and devices. Network interface subsystem 830 serves as an interface for receiving data from and transmitting data to other systems from computer system 800.

User interface input devices 820 may include a keyboard, pointing devices such as a mouse, trackball, touchpad, or graphics tablet, a scanner, a barcode scanner, a touchscreen incorporated into the display, audio input devices such as voice recognition systems, microphones, and other types of input devices. In general, use of the term "input device" is intended to include all possible types of devices and mechanisms for inputting information to computer system 800.

User interface output devices 825 may include a display subsystem, a printer, a fax machine, or non-visual displays such as audio output devices, etc. The display subsystem may be a cathode ray tube (CRT), a flat-panel device such as a liquid crystal display (LCD), or a projection device. In general, use of the term "output device" is intended to include all possible types of devices and mechanisms for outputting information from computer system 800.

Storage subsystem 815 may be configured to store the basic programming and data constructs that provide the functionality of the present invention. Software (code modules or instructions) that provides the functionality of the present invention may be stored in storage subsystem 815. These software modules or instructions may be executed by processor(s) 805. Storage subsystem 815 may also provide a repository for storing data used in accordance with the present invention. Storage subsystem 815 may comprise memory subsystem 835 and file/disk storage subsystem 855.
Memory subsystem 835 may include a number of memories including a main random access memory (RAM) 850 for storage of instructions and data during program execution and a read only memory (ROM) 840 in which fixed instructions are stored. File storage subsystem 855 provides persistent (non-volatile) storage for program and data files, and may include a hard disk drive, a floppy disk drive along with associated removable media, a Compact Disk Read Only Memory (CD-ROM) drive, a DVD, an optical drive, removable media cartridges, and other like storage media.

Computer system 800 can be of various types including a personal computer, a portable computer, a workstation, a network computer, a mainframe, a kiosk, or any other data processing system. Due to the ever-changing nature of computers and networks, the description of computer system 800 depicted in FIG. 8 is intended only as a specific example for purposes of illustrating the preferred embodiment of the computer system. Many other configurations having more or fewer components than the system depicted in FIG. 8 are possible.

Although specific embodiments of the invention have been described, various modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the invention. The described invention is not restricted to operation within certain specific data processing environments, but is free to operate within a plurality of data processing environments. Additionally, although the present invention has been described using a particular series of transactions and steps, it should be apparent to those skilled in the art that the scope of the present invention is not limited to the described series of transactions and steps.

Further, while the present invention has been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the present invention. The present invention may be implemented only in hardware, or only in software, or using combinations thereof.

The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that additions, subtractions, deletions, and other modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the claims.
WHAT IS CLAIMED IS:

1. A method for providing controlled key management, the method comprising:
   generating a request indicative of a key management function;
   receiving the request at a first of a number of intermediate parties capable of
   relaying the request toward a key management server;
   performing the key management function subsequent to receiving the request
   from a last of the intermediate parties that is authorized to provide the request to the key
   management server; and
   generating a response to the request.

2. The method of claim 1 wherein receiving the request at the first of a
   number of intermediate parties capable of relaying the request toward a key management
   server comprises receiving the request at the first of the number of intermediate parties using
   a store and forward protocol.

3. The method of claim 1 wherein receiving the request at the first of a
   number of intermediate parties capable of relaying the request toward a key management
   server comprises receiving the request at an e-mail server.

4. The method of claim 1 wherein the number of intermediate parties is
   one such that receiving the request at the first of a number of intermediate parties capable of
   relaying the request toward a key management server comprises receiving the request at the
   last of the number of intermediate parties.

5. The method of claim 1 further comprising:
   receiving the response to the request at the first of a number of intermediate
   parties capable of relaying the response to a client; and
   wherein the last of the number of intermediate parties is configured to deliver
   the response to the client.

6. The method of claim 1 wherein the key management function
   comprises at least one of a create operation, a store operation, a retrieve operation, a find
   operation, a disable operation, a destroy operation, and a modify operation.
7. The method of claim 1 further comprising:
   encrypting the request; and
   digitally signing the request.

8. The method of claim 7 further comprising:
   authenticating the request using the digital signature; and
   decrypting the request based on a positive determination that the request is
   authentic.

9. The method of claim 1 further comprising:
   encrypting the response; and
   digitally signing the response.

10. The method of claim 9 further comprising:
    authenticating the response using the digital signature; and
    decrypting the response based on a positive determination that the request is
    authentic.

11. A computer program product stored on a computer readable medium
    for providing controlled key management, the computer program product comprising:
    code for generating a request indicative of a key management function;
    code for receiving the request at a first of a number of intermediate parties
    capable of relaying the request toward a key management server;
    code for performing the key management function subsequent to receiving the
    request from a last of the intermediate parties that is authorized to provide the request to the
    key management server; and
    code for generating a response to the request.

12. The computer program product of claim 11 wherein the code for
    receiving the request at the first of a number of intermediate parties capable of relaying the
    request toward a key management server comprises code for receiving the request at the first
    of the number of intermediate parties using a store and forward protocol.
13. The computer program product of claim 11 wherein the code receiving
the request at the first of a number of intermediate parties capable of relaying the request
toward a key management server comprises code for receiving the request at an e-mail server.

14. The computer program product of claim 11 wherein the number of
intermediate parties is one such that the code for receiving the request at the first of a number
of intermediate parties capable of relaying the request toward a key management server
comprises code for receiving the request at the last of the number of intermediate parties.

15. The computer program product of claim 11 further comprising:
   code for receiving the response to the request at the first of a number of
   intermediate parties capable of relaying the response to a client; and
   wherein the last of the number of intermediate parties is configured to deliver
   the response to the client.

16. The computer program product of claim 11 wherein the key
management function comprises at least one of a create operation, a store operation, a retrieve
operation, a find operation, a disable operation, a destroy operation, and a modify operation.

17. The computer program product of claim 11 further comprising:
   code for encrypting the request; and
   code for digitally signing the request.

18. The computer program product of claim 17 further comprising:
   code for authenticating the request using the digital signature; and
   code for decrypting the request based on a positive determination that the
   request is authentic.

19. The computer program product of claim 11 further comprising:
   code for encrypting the response; and
   code for digitally signing the response.

20. The method of claim 19 further comprising:
   code for authenticating the response using the digital signature; and
   code for decrypting the response based on a positive determination that the
   request is authentic.
21. A system for providing controlled key management, the system comprising:
a number of intermediate parties, wherein a first of the number of intermediate
parties is configured to receive a request indicative of a key management function from a
client, and wherein a last of the number of intermediate parties is authorized to provide the
request to one or more key management servers; and
one or more key management servers configured to:
receive the request from the last of the number of intermediate parties;
perform the key management function; and
generate a response to the request.

22. A system for secured key management, the system comprising:
a key management server; and
a first server communicatively positioned between the key management server
and a client and configured to:
receive a request addressed to the key management server from the
client, the request indicative of a key management function;
deliver the request to the key management server if the first server is
authorized to deliver a request from a client to the key management server, and
relay the request to a second server communicatively positioned
between the key management server and the client if the first server is not configured to
access the key management server.

23. The system of claim 22 wherein the first server is further configured to
receive the request using a store and forward protocol.

24. The system of claim 22 wherein the first server comprises an e-mail
server.

25. The system of claim 22 wherein the first server is further configured
to:
receive a response to the request addressed to the client;
deliver the response to the client if the first server is configured to access the
client; and
26. A system for providing controlled key management, the system comprising:
   a processor; and
   a memory coupled to the processor, the memory configured to store a plurality
   of code modules which when executed by the processor cause the processor to:
   receive a request addressed to a key management server from a client,
   the request indicative of a key management function;
   deliver the request to the key management server based on receiving
   authorization to deliver requests from clients to the key management server; and
   relay the request to a first host communicatively positioned between
   the key management server and the client if not permitted to directly access the key
   management server.

27. The system of claim 26 wherein the processor is configured to receive
    the request using a store and forward protocol.

28. The system of claim 26 wherein the processor is configured to receive
    the request using a simple mail transfer protocol.

29. The system of claim 26 wherein the processor is further configured to:
    receive a response to the request addressed to the client;
    deliver the response to the client if capable of accessing the client; and
    relay the response to a second host communicatively positioned between the
    key management server and the client if not capable of directly accessing the client.
FIG. 1
(PRIOR ART)
BEGIN

GENERATE REQUEST INDICATIVE OF KEY MANAGEMENT FUNCTION

RECEIVE REQUEST AT FIRST OF NUMBER OF INTERMEDIATE PARTIES CAPABLE OF RELAYING REQUEST TOWARD KEY MANAGEMENT SERVER

INTERMEDIATE PARTY AUTHORIZED TO PROVIDE REQUEST TO KEY MANAGEMENT SERVER?

YES

NO

RECEIVE REQUEST AT NEXT INTERMEDIATE PARTY CAPABLE OF RELAYING REQUEST TOWARD KEY MANAGEMENT SERVER

RECEIVE REQUEST FROM LAST OF INTERMEDIATE PARTIES

FIG. 3A
PERFORM KEY MANAGEMENT FUNCTION

GENERATE RESPONSE INDICATIVE OF PERFORMANCE OF KEY MANAGEMENT FUNCTION

RECEIVE RESPONSE AT FIRST OF NUMBER OF INTERMEDIATE PARTIES CAPABLE OF RELAYING RESPONSE TOWARD CLIENT

INTERMEDIATE PARTY CAPABLE OF DELIVERING RESPONSE TO CLIENT?

YES

RECEIVE REQUEST AT NEXT INTERMEDIATE PARTY CAPABLE OF RELAYING RESPONSE TOWARD CLIENT

NO

RECEIVE RESPONSE FROM LAST OF INTERMEDIATE PARTIES

END

FIG. 3B
FIG. 4
700 KMS 650
722 A
724 DETERMINE WHETHER AUTHORIZED TO DELIVER REQUEST TO KMS 650
726 NO
728 AUTHORIZED?
730 STORE REQUEST IN MAILBOX FOR KMS 650
732 AUTHENTICATE TO SMTP SERVER 650 AND RETRIEVE REQUESTS FROM MAILBOX
734 B

FIG. 7B
700 KMS 650

736 PROCESS DIGITAL SIGNATURE TO AUTHENTICATE REQUEST

738

740 AUTHENTIC? NO

742 IGNORE REQUEST

YES

744 DECRYPT REQUEST

746 PERFORM KEY MANAGEMENT FUNCTION

748 GENERATE RESPONSE INDICATING RESULTS OF KEY MANAGEMENT FUNCTION

750 ENCRYPT AND DIGITALLY SIGN RESPONSE

752 DELIVER RESPONSE TO E-MAIL SERVER 640 ADDRESSED TO CLIENT 610

754 FORWARD RESPONSE TO E-MAIL SERVER 620

756 STORE RESPONSE IN MAILBOX FOR CLIENT 610

758 RETRIEVE RESPONSE FROM E-MAIL SERVER 620

760 PROCESS RESPONSE TO OBTAIN RESULTS OF KEY MANAGEMENT FUNCTION

FIG. 7C
INTERNATIONAL SEARCH REPORT

PCT/ISA210 (second sheet) (My 2008)

INTERNATIONAL SEARCH REPORT

International application No

PCT/ISA/210 (second sheet) (My 2008)

A. CLASSIFICATION OF SUBJECT MATTER

H04L 9/00(2006.01)i, H04L 9/32(2006.01)i

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)

IPC 8 H04L 9/

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

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and applications for Utility models since 1975

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

eKIPASS(KIPO internal) "key, management, sever, client, intermediate, protocol, request, e-mail"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 6,993,652 B2 (Alexander Medvinsky) 31 JAN 2006 See abstract and figures 1-2</td>
<td>1 - 29</td>
</tr>
<tr>
<td>A</td>
<td>JP 2006-148469 A (Fujitsu Ltd) 8 JUN 2006 See abstract and figures 3-5</td>
<td>1 - 29</td>
</tr>
<tr>
<td>A</td>
<td>US 2004/0268152 A1 (Sharon Xia et al) 20 DEC 2004 See abstract and figure 5</td>
<td>1 - 29</td>
</tr>
</tbody>
</table>

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 but published on or after the international filing date
  *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of 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

Date of the actual completion of the international search
30 SEPTEMBER 2008 (30 09 2008)

Date of mailing of the international search report
30 SEPTEMBER 2008 (30.09.2008)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No 82-42-472-7140

Authorized officer

MA, Jung Youn

Telephone No 82-42-481-5679

Form PCT/ISA/210 (second sheet) (My 2008)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2005-050328 A1</td>
<td>03.03.2005</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA 2463034 A1</td>
<td>17.04.2003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 1611031 A</td>
<td>27.04.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2005-505991 T2</td>
<td>24.02.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 10-2004-0045486</td>
<td>01.06.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PA 04003226 A</td>
<td>08.07.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 6993652 BB</td>
<td>31.01.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 0303-2575A2</td>
<td>17.04.2003</td>
</tr>
<tr>
<td>JP 2006-148469 A</td>
<td>08.06.2006</td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>