

The system compares the two hashes resulting from the immediately preceding steps. If they match, the image in memory has not been tampered with.

Application Interface Functions

As described above, interfaces are exposed to application programs so that application programs can take advantage of protected storage features without having to implement sophisticated encryption schemes and without having to make RPC calls. These interfaces and their functions are described in an attached appendix that forms part of this document. The appendix also provides explanations regarding the proper usage of the interfaces.

Conclusion

The invention provides a versatile and efficient architecture that provides a number of advantages over the prior art. One significant advantage is that different application programs can utilize a single, provided server to store core data secrets in a central storage area. This promotes consistency among the applications and removes significant overhead from the applications. The user interface is one area that benefits from the consistency provided by the storage system described above, since user prompts are generated by the system rather than by the individual application programs. Storing data items in a uniform manner also allows them to be managed by a single management program that is independent of the application programs themselves.

Another significant advantage of the invention is that the underlying details of securing data items are hidden from calling application programs. Thus, program developers do not have to implement sophisticated security measures; such measures can be implemented with simple calls to the storage system described herein. An added benefit is that new technologies such as smart cards will be available to application programs without extensive reprogramming.

The invention protects secrets from user-oriented and software-oriented attacks, including attacks from viruses. Significantly, access control is managed outside the application programs that generate and access data items. Because applications do not have direct access to keying material or other control data, access to one piece of data does not imply access to any other data. Furthermore, the storage system itself does not retain the information required to decrypt stored data items. Rather, the user must be present and must supply a correct password to allow data decryption.

A further important benefit of the invention is that users are not forced to explicitly enter passwords when data access is required. Rather, user authentication is performed once, when the user logs on to the computer or network. This logon information is used for both user authentication and to derive keys for data encryption and decryption. Steps are taken to backup or escrow keys derived from logon information, so that such keys can be recovered when the user's password changes without notification to the storage system. However, recovery of escrowed keys requires a successful network logon.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as exemplary forms of implementing the claimed invention.

We claim:

1. A method of securely storing and recovering data protection keys, comprising the following steps:
 - deriving a client key from a user secret that is supplied by a user during network logon procedures;
 - securing user data on a client computer with the client key;
 - sending the client key to a network supervisory computer that authenticates network users during user logon procedures;
 - encrypting a data combination at the network supervisory computer, the data combination including the client key and a user identification corresponding to a currently authenticated current user of the client computer;
 - returning the encrypted data combination to the client computer;
 - storing the encrypted data combination at the client computer;
 - sending the encrypted data combination to the network supervisory computer in order to recover the client key;
 - decrypting the data combination at the network supervisory computer to obtain the client key and the user identification in response to receiving the encrypted data combination from the client computer;
 - returning the client key to the client computer only if the obtained user identification corresponds to the currently authenticated user of the client computer.
2. A method as recited in claim 1, further comprising an additional step of encrypting the client key before sending it to the network supervisory computer.
3. A method as recited in claim 1, further comprising the following additional steps:
 - encrypting the client key before sending it to the network supervisory computer;
 - decrypting the client key at the client computer after returning the client key to the client computer.
4. A method as recited in claim 1, further comprising an additional step of creating a message authentication code based at least in part on the client key and the user identification, the data combination further including the message authentication code.
5. A method as recited in claim 1, further comprising an additional step of creating a message authentication code based at least in part on the client key and the user identification, the data combination further including the message authentication code, wherein the step of returning the client key is conditioned upon a successful authentication of the client key and the user identification using the message authentication code.
6. A method as recited in claim 1, wherein the encrypting step is performed using an encryption key that is derived from a master key and a random key, the method further comprising the following additional steps:
 - returning the random key with the encrypted data combination to the client computer; and
 - storing the random key at the client computer.
7. A method as recited in claim 1, wherein the encrypting step is performed using an encryption key that is derived from a master key and a random key, the method further comprising the following additional steps:
 - returning the random key with the encrypted data combination to the client computer;
 - storing the random key at the client computer;
 - sending the random key to the network supervisory computer from the client computer in order to recover the client key.