- **Authorization:** Granting access to specific services and/or resources based on the authentication.
- **Provisioning:** Includes creating an account in each target system for the user, enrollment or registration of user in accounts, establishment of access rights or credentials to ensure the privacy and integrity of account data.
- **Management:** Services related to runtime configuration and deployment.

Note that Kerberos contains a number of elements of an identity management system.

Figure 4.8 [LINN06] illustrates entities and data flows in a generic identity management architecture. A **principal** is an identity holder. Typically, this is a human user that seeks access to resources and services on the network. User devices, agent processes, and server systems may also function as principals. Principals authenticate themselves to an **identity provider**. The identity provider associates authentication information with a principal, as well as attributes and one or more identifiers.

Increasingly, digital identities incorporate attributes other than simply an identifier and authentication information (such as passwords and biometric information). An **attribute service** manages the creation and maintenance of such attributes. For example, a user needs to provide a shipping address each time an order is placed at a new Web merchant, and this information needs to be revised when the user moves. Identity management enables the user to provide this information once, so that it is maintained in a single place and released to data consumers in accordance with authorization and privacy policies. Users may create some of the attributes to be associated with their digital identity, such as address. **Administrators** may also assign attributes to users, such as roles, access permissions, and employee information.

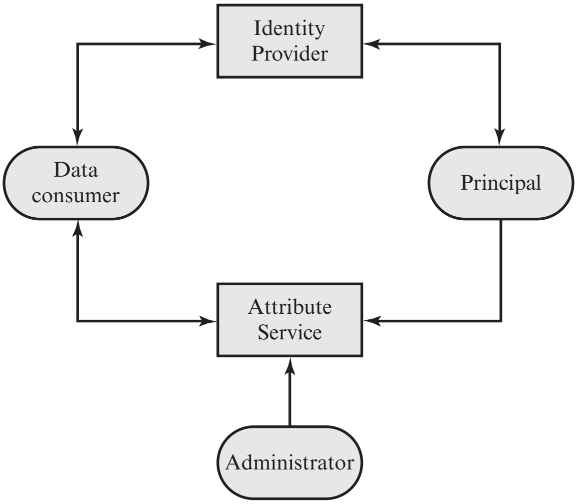


Figure 4.8 Generic Identity Management Architecture

Data consumers are entities that obtain and employ data maintained and provided by identity and attribute providers, which are often used to support authorization decisions and to collect audit information. For example, a database server or file server is a data consumer that needs a client's credentials so as to know what access to provide to that client.

Identity Federation

Identity federation is, in essence, an extension of identity management to multiple security domains. Such domains include autonomous internal business units, external business partners, and other third-party applications and services. The goal is to provide the sharing of digital identities so that a user can be authenticated a single time and then access applications and resources across multiple domains. Because these domains are relatively autonomous or independent, no centralized control is possible. Rather, the cooperating organizations must form a federation based on agreed standards and mutual levels of trust to securely share digital identities.

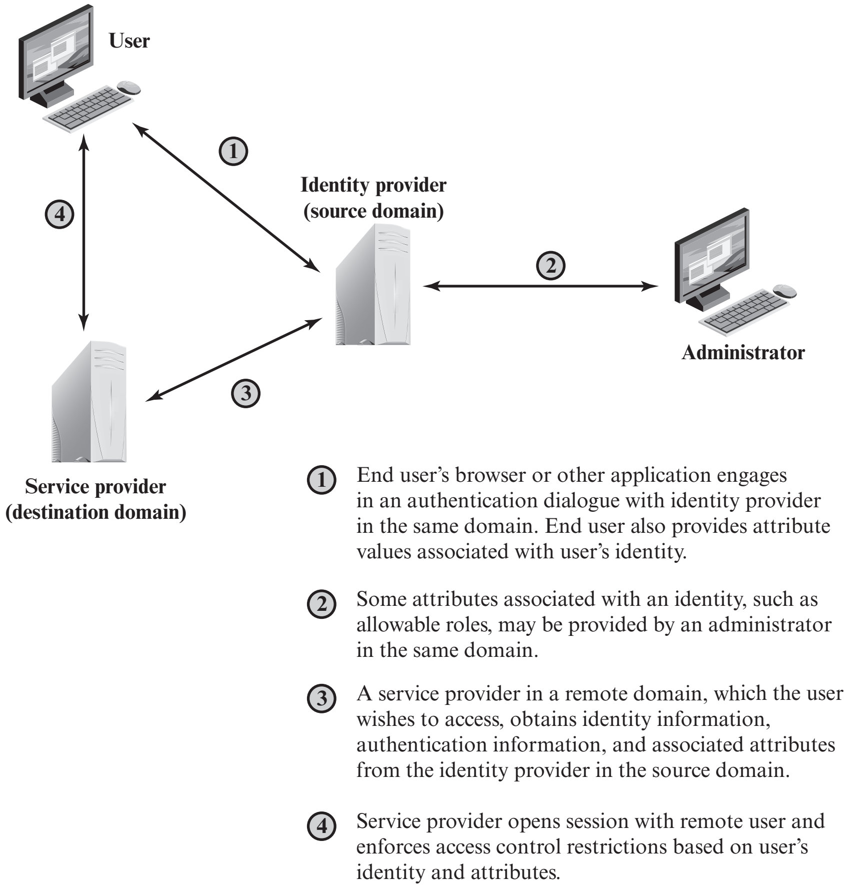
Federated identity management refers to the agreements, standards, and technologies that enable the portability of identities, identity attributes, and entitlements across multiple enterprises and numerous applications and supports many thousands, even millions, of users. When multiple organizations implement interoperable federated identity schemes, an employee in one organization can use a single sign-on to access services across the federation with trust relationships associated with the identity. For example, an employee may log onto her corporate intranet and be authenticated to perform authorized functions and access authorized services on that intranet. The employee could then access her health benefits from an outside health-care provider without having to reauthenticate.

Beyond SSO, federated identity management provides other capabilities. One is a standardized means of representing attributes. Increasingly, digital identities incorporate attributes other than simply an identifier and authentication information (such as passwords and biometric information). Examples of attributes include account numbers, organizational roles, physical location, and file ownership. A user may have multiple identifiers; for example, each identifier may be associated with a unique role with its own access permissions.

Another key function of federated identity management is identity mapping. Different security domains may represent identities and attributes differently. Furthermore, the amount of information associated with an individual in one domain may be more than is necessary in another domain. The federated identity management protocols map identities and attributes of a user in one domain to the requirements of another domain.

Figure 4.9 illustrates entities and data flows in a generic federated identity management architecture.

The identity provider acquires attribute information through dialogue and protocol exchanges with users and administrators. For example, a user needs to provide a shipping address each time an order is placed at a new Web merchant, and this information needs to be revised when the user moves. Identity management enables the user to provide this information once, so that it is maintained in a



1. End user's browser or other application engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identity.

2. Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.

3. A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information, and associated attributes from the identity provider in the source domain.

4. Service provider opens session with remote user and enforces access control restrictions based on user's identity and attributes.

Figure 4.9 Federated Identity Operation

single place and released to data consumers in accordance with authorization and privacy policies.

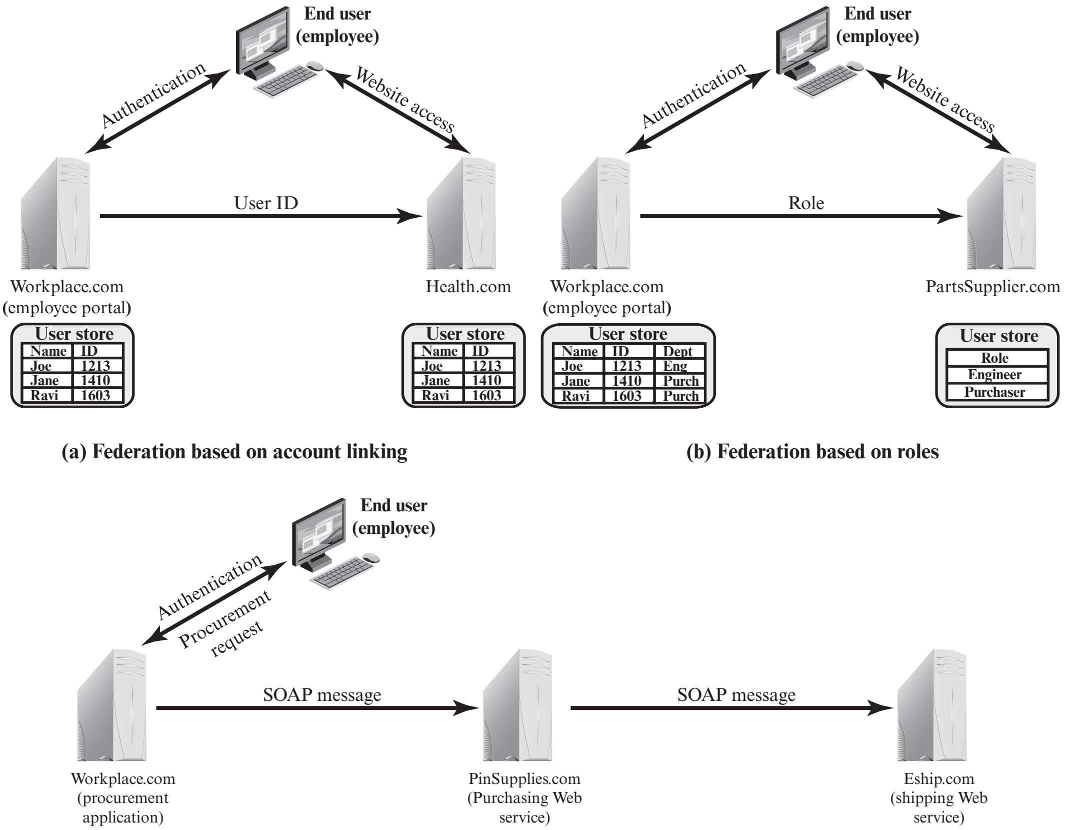
Service providers are entities that obtain and employ data maintained and provided by identity providers, often to support authorization decisions and to collect audit information. For example, a database server or file server is a data consumer that needs a client's credentials so as to know what access to provide to that client. A service provider can be in the same domain as the user and the identity provider. The power of this approach is for federated identity management, in which the service provider is in a different domain (e.g., a vendor or supplier network).

Standards Federated identity management uses a number of standards as the building blocks for secure identity exchange across different domains or heterogeneous systems. In essence, organizations issue some form of security tickets for their users that can be processed by cooperating partners. Identity federation standards are thus concerned with defining these tickets in terms of content and format, providing protocols for exchanging tickets, and performing a number of management tasks. These tasks include configuring systems to perform attribute transfers and identity mapping, and performing logging and auditing functions. The key standards are as follows:

- **The Extensible Markup Language (XML):** A markup language uses sets of embedded tags or labels to characterize text elements within a document so as to indicate their appearance, function, meaning, or context. XML documents appear similar to HTML (Hypertext Markup Language) documents that are visible as Web pages, but provide greater functionality. XML includes strict definitions of the data type of each field, thus supporting database formats and semantics. XML provides encoding rules for commands that are used to transfer and update data objects.
- **The Simple Object Access Protocol (SOAP):** A minimal set of conventions for invoking code using XML over HTTP. It enables applications to request services from one another with XML-based requests and receive responses as data formatted with XML. Thus, XML defines data objects and structures, and SOAP provides a means of exchanging such data objects and performing remote procedure calls related to these objects. See [ROS06] for an informative discussion.
- **WS-Security:** A set of SOAP extensions for implementing message integrity and confidentiality in Web services. To provide for secure exchange of SOAP messages among applications, WS-Security assigns security tokens to each message for use in authentication.
- **Security Assertion Markup Language (SAML):** An XML-based language for the exchange of security information between online business partners. SAML conveys authentication information in the form of assertions about subjects. Assertions are statements about the subject issued by an authoritative entity.

The challenge with federated identity management is to integrate multiple technologies, standards, and services to provide a secure, user-friendly utility. The key, as in most areas of security and networking, is the reliance on a few mature standards widely accepted by industry. Federated identity management seems to have reached this level of maturity.

Examples To get some feel for the functionality of identity federation, we look at three scenarios, taken from [COMP06]. In the first scenario (Figure 4.10a), Workplace.com contracts with Health.com to provide employee health benefits. An employee uses a Web interface to sign on to Workplace.com and goes through an authentication procedure there. This enables the employee to access authorized services and resources at Workplace.com. When the employee clicks on a link to access health benefits, her browser is redirected to Health.com. At the same time, the Workplace.com software passes the user's identifier to Health.com in a secure manner. The two organizations are part of a federation that cooperatively exchanges user identifiers. Health.com maintains user identities for every employee at Workplace.com and associates with each identity health-benefits information



User store

Name	ID
Joe	1213
Jane	1410
Ravi	1603

User store

Name	ID
Joe	1213
Jane	1410
Ravi	1603

(a) Federation based on account linking

User store

Name	ID	Dept
Joe	1213	Eng
Jane	1410	Purch
Ravi	1603	Purch

User store

Role
Engineer
Purchaser

(b) Federation based on roles

(c) Chained Web Services

Figure 4.10 Federated Identity Scenarios

and access rights. In this example, the linkage between the two companies is based on account information and user participation is browser based.

Figure 4.10b shows a second type of browser-based scheme. PartsSupplier.com is a regular supplier of parts to Workplace.com. In this case, a role-based access control (RBAC) scheme is used for access to information. An engineer of Workplace.com authenticates at the employee portal at Workplace.com and clicks on a link to access information at PartsSupplier.com. Because the user is authenticated in the role of an engineer, he is taken to the technical documentation and troubleshooting portion of PartsSupplier.com's Web site without having to sign on. Similarly, an employee in a purchasing role signs on at Workplace.com and is authorized, in that role, to place purchases at PartsSupplier.com without having to authenticate to PartsSupplier.com. For this scenario, PartsSupplier.com does not have identity information for individual employees at Workplace.com. Rather, the linkage between the two federated partners is in terms of roles.

The scenario illustrated in Figure 4.10c can be referred to as document based rather than browser based. In this third example, Workplace.com has a purchasing agreement with PinSupplies.com, and PinSupplies.com has a business relationship with E-Ship.com. An employee of Workplace.com signs on and is authenticated to make purchases. The employee goes to a procurement application that provides a list of Workplace.com's suppliers and the parts that can be ordered. The user clicks on the PinSupplies button and is presented with a purchase order Web page (HTML page). The employee fills out the form and clicks the submit button. The procurement application generates an XML/SOAP document that it inserts into the envelope body of an XML-based message. The procurement application then inserts the user's credentials in the envelope header of the message, together with Workplace.com's organizational identity. The procurement application posts the message to the PinSupplies.com's purchasing Web service. This service authenticates the incoming message and processes the request. The purchasing Web service then sends a SOAP message its shipping partner to fulfill the order. The message includes a PinSupplies.com security token in the envelope header and the list of items to be shipped as well as the end user's shipping information in the envelope body. The shipping Web service authenticates the request and processes the shipment order.

4.8 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

authentication
authentication server (AS)
federated identity management
identity management
Kerberos
Kerberos realm
key distribution
key distribution center (KDC)
key management
master key
mutual authentication
nonce
one-way authentication
propagating cipher block chaining (PCBC) mode
public-key certificate
public-key directory
realm
replay attack
ticket
ticket-granting server (TGS)
timestamp
X.509 certificate

Review Questions

4.1 Explain the operation of a key distribution center.

4.2 What are the advantages of the automated key distribution approach?

4.3 What is Kerberos?

4.4 Identify the security threats that exist in an open distributed network.

4.5 In the context of Kerberos, what is a realm?

4.6 What are the ingredients of a authentication server's ticket? Explain the significance of each.

4.7 List the environmental shortcomings and technical deficiencies of Kerberos version 4.

4.8 Identify the weakness of a public key distribution with a public key algorithm. How can it be fixed?

4.9 What is a X.509 certificate?

4.10 Explain the different fields of the public-key certificate of the X.509 scheme.

4.11 What is public-key infrastructure?

4.12 What are the key elements of the PKIX model?

4.13 Name the PKIX certificate management protocols.

4.14 What is federated identity management?

Problems

4.1 "We are under great pressure, Holmes." Detective Lestrade looked nervous. "We have learned that copies of sensitive government documents are stored in computers of one foreign embassy here in London. Normally these documents exist in electronic form only on a selected few government computers that satisfy the most stringent security requirements. However, sometimes they must be sent through the network connecting all government computers. But all messages in this network are encrypted using a top secret encryption algorithm certified by our best crypto experts. Even the NSA and the KGB are unable to break it. And now these documents have appeared in hands of diplomats of a small, otherwise insignificant, country. And we have no idea how it could happen."

"But you do have some suspicion who did it, do you?" asked Holmes.

"Yes, we did some routine investigation. There is a man who has legal access to one of the government computers and has frequent contacts with diplomats from the embassy. But the computer he has access to is not one of the trusted ones where these documents are normally stored. He is the suspect, but we have no idea how he could obtain copies of the documents. Even if he could obtain a copy of an encrypted document, he couldn't decrypt it."

"Hmm, please describe the communication protocol used on the network." Holmes opened his eyes, thus proving that he had followed Lestrade's talk with an attention that contrasted with his sleepy look.

"Well, the protocol is as follows. Each node N of the network has been assigned a unique secret key K_n. This key is used to secure communication between the node and a trusted server. That is, all the keys are stored also on the server. User A, wishing to send a secret message M to user B, initiates the following protocol:

1. A generates a random number R and sends to the server his name A, destination B, and $E(K_a, R)$.
2. Server responds by sending $E(K_b, R)$ to A.
3. A sends $E(R, M)$ together with $E(K_b, R)$ to B.
4. B knows K_b, thus decrypts $E(K_b, R)$ to get R and will subsequently use R to decrypt $E(R, M)$ to get M.

You see that a random key is generated every time a message has to be sent. I admit the man could intercept messages sent between the top secret trusted nodes, but I see no way he could decrypt them."

"Well, I think you have your man, Lestrade. The protocol isn't secure because the server doesn't authenticate users who send him a request. Apparently designers of the protocol have believed that sending $E(K_x, R)$ implicitly authenticates user X as the sender, as only X (and the server) knows K_x. But you know that $E(K_x, R)$ can be intercepted and later replayed. Once you understand where the hole is, you will be able to obtain enough evidence by monitoring the man's use of the computer he has access to. Most likely he works as follows: After intercepting $E(K_a, R)$ and $E(R, M)$ (see steps 1 and 3 of the protocol), the man, let's denote him as Z, will continue by pretending to be A and . . .

Finish the sentence for Holmes.

4.2 There are three typical ways to use nonces as challenges. Suppose N_a is a nonce generated by A, A and B share key K, and f() is a function (such as increment). The three usages are

Usage 1	Usage 2	Usage 3
(1) A → B: N_a	(1) A → B: E(K, N_a)	(1) A → B: E(K, N_a)
(2) B → A: E(K, N_a)	(2) B → A: N_a	(2) B → A: E(K, f(N_a))

Describe situations for which each usage is appropriate.

4.3 Show that a random error in one block of ciphertext is propagated to all subsequent blocks of plaintext in PCBC mode (see Figure F.2 in Appendix F).

4.4 Suppose that, in PCBC mode, blocks C_i and C_{i+1} are interchanged during transmission. Show that this affects only the decrypted blocks P_i and P_{i+1} but not subsequent blocks.

4.5 In addition to providing a standard for public-key certificate formats, X.509 specifies an authentication protocol. The original version of X.509 contains a security flaw. The essence of the protocol is

$$\begin{aligned} &A \rightarrow B: \quad A\,\{t_A, r_A, ID_B\} \\ &B \rightarrow A: \quad B\,\{t_B, r_B, ID_A, r_A\} \\ &A \rightarrow B: \quad A\,\{r_B\} \end{aligned}$$

where t_A and t_B are timestamps, r_A and r_B are nonces, and the notation X {Y} indicates that the message Y is transmitted, encrypted, and signed by X.

The text of X.509 states that checking timestamps t_A and t_B is optional for three-way authentication. But consider the following example: Suppose A and B have used the preceding protocol on some previous occasion, and that opponent C has intercepted the preceding three messages. In addition, suppose that timestamps are not used and are all set to 0. Finally, suppose C wishes to impersonate A to B. C initially sends the first captured message to B:

$$C \rightarrow B: \quad A\,\{0, r_A, ID_B\}$$

B responds, thinking it is talking to A but is actually talking to C:

$$B \rightarrow C: \quad B\,\{0, r'_B, ID_A, r_A\}$$

C meanwhile causes A to initiate authentication with C by some means. As a result, A sends C the following:

$$A \rightarrow C: \quad A\,\{0, r'_A, ID_C\}$$

C responds to A using the same nonce provided to C by B.

$$C \rightarrow A: \quad C\,\{0, r'_B, ID_A, r'_A\}$$

A responds with

$$A \rightarrow C: \quad A\,\{r'_B\}$$

This is exactly what C needs to convince B that it is talking to A, so C now repeats the incoming message back out to B.

$$C \rightarrow B: \quad A\,\{r'_B\}$$

So B will believe it is talking to A, whereas it is actually talking to C. Suggest a simple solution to this problem that does not involve the use of timestamps.

4.6 Consider a one-way authentication technique based on asymmetric encryption:

$$\begin{aligned} &\mathrm{A} \rightarrow \mathrm{B}: && ID_A \\ &\mathrm{B} \rightarrow \mathrm{A}: && R_1 \\ &\mathrm{A} \rightarrow \mathrm{B}: && \mathrm{E}(PR_a, R_1) \end{aligned}$$

a. Explain the protocol.
b. What type of attack is this protocol susceptible to?

4.7 Consider a one-way authentication technique based on asymmetric encryption:

$$\begin{aligned} &\mathrm{A} \rightarrow \mathrm{B}: && ID_A \\ &\mathrm{B} \rightarrow \mathrm{A}: && \mathrm{E}(\mathrm{PU}_a, R_2) \\ &\mathrm{A} \rightarrow \mathrm{B}: && R_2 \end{aligned}$$

a. Explain the protocol.
b. What type of attack is this protocol susceptible to?

4.8 In Kerberos, how do servers verify the authenticity of the client using the ticket?

4.9 In Kerberos, how does an authentication server protect a ticket from being altered by the client or opponent?

4.10 How is ticket reuse by an opponent prevented in Kerberos?

4.11 What is the purpose of a session key in Kerberos? How is it distributed by the AS?

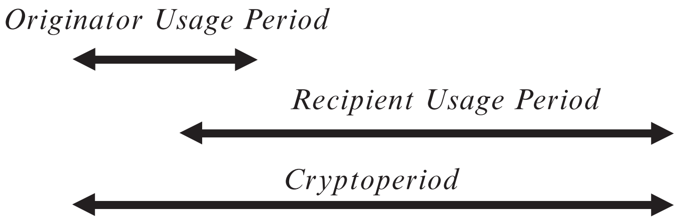
4.12 The 1988 version of X.509 lists properties that RSA keys must satisfy to be secure, given current knowledge about the difficulty of factoring large numbers. The discussion concludes with a constraint on the public exponent and the modulus n:

> It must be ensured that $e > \log_2(n)$ to prevent attack by taking the eth root mod n to disclose the plaintext.

Although the constraint is correct, the reason given for requiring it is incorrect. What is wrong with the reason given and what is the correct reason?

4.13 Find at least one intermediate certification authority's certificate and one trusted root certification authority's certificate on your computer (e.g., in the browser). Print screenshots of both the general and details tab for each certificate.

4.14 NIST defines the term "cryptoperiod" as the time span during which a specific key is authorized for use or in which the keys for a given system or application may remain in effect. One document on key management uses the following time diagram for a shared secret key.



Explain the overlap by giving an example application in which the originator's usage period for the shared secret key begins before the recipient's usage period and also ends before the recipient's usage period.

4.15 Consider the following protocol, designed to let A and B decide on a fresh, shared session key K'_{AB}. We assume that they already share a long-term key K_{AB}.

1. $A \rightarrow$ B: A, N_A
2. $B \rightarrow$ A: $E(K_{AB}, [N_A, K'_{AB}])$
3. $A \rightarrow B$: $E(K'_{AB}, N_A)$

   a. We first try to understand the protocol designer's reasoning:
   - Why would A and B believe after the protocol ran that they share K'_{AB} with the other party?
   - Why would they believe that this shared key is fresh?

   In both cases, you should explain both the reasons of both A and B, so your answer should complete the following sentences.

   A believes that she shares K'_{AB} with B since . . .
   B believes that he shares K'_{AB} with A since . . .
   A believes that K'_{AB} is fresh since . . .
   B believes that K'_{AB} is fresh since . . .

   b. Assume now that A starts a run of this protocol with B. However, the connection is intercepted by the adversary C. Show how C can start a new run of the protocol using reflection, causing A to believe that she has agreed on a fresh key with B (in spite of the fact that she has only been communicating with C). Thus, in particular, the belief in (a) is false.

   c. Propose a modification of the protocol that prevents this attack.

4.16 List the different management functions of the PKIX model.

4.17 Explain the entities and data flows of generic identity management architecture.

4.18 Consider the following protocol:

$$\begin{array}{ll} \text{A} \rightarrow \text{KDC:} & ID_A \| ID_B \| N_1 \\ \text{KDC} \rightarrow \text{A:} & \text{E}(K_a, [K_S \| ID_B \| N_1 \| \text{E}(K_b, [K_S \| ID_A])) \\ \text{A} \rightarrow \text{B:} & \text{E}(K_b, [K_S \| ID_A]) \\ \text{B} \rightarrow \text{A:} & \text{E}(K_S, N_2)\text{A} \rightarrow \text{B:} \quad \text{E}(K_S, \text{f}(N_2)) \end{array}$$

a. Explain the protocol.
b. Can you think of a possible attack on this protocol? Explain how it can be done.
c. Mention a possible technique to get around the attack—not a detailed mechanism, just the basics of the idea.

Note: The remaining problems deal with a cryptographic product developed by IBM, which is briefly described in a document at this book's Web site in `IBMCrypto.pdf`*. Try these problems after reviewing the document.*

4.19 What is the effect of adding the instruction EMK$_i$?

$$\text{EMK}_i\text{: } X \rightarrow \text{E}(KMH_i, X) \quad i = 0, 1$$

4.20 Suppose N different systems use the IBM Cryptographic Subsystem with host master keys KMH$[i](i = 1, 2, \ldots, N)$. Devise a method for communicating between systems without requiring the system to either share a common host master key or to divulge their individual host master keys. *Hint*: Each system needs three variants of its host master key.

4.21 The principal objective of the IBM Cryptographic Subsystem is to protect transmissions between a terminal and the processing system. Devise a procedure, perhaps adding instructions, which will allow the processor to generate a session key KS and distribute it to Terminal i and Terminal j without having to store a key-equivalent variable in the host.