1. What are the main objectives in designing a secure database application? Explain the terms, security, integrity, availability, and authentication. (Section 21.1)

Answer:

The main objectives in designing a secure database application are *security, integrity* and *availability.*

Secrecy: Information should not be disclosed to unauthorized users. For example, a student should not be allowed to examine other students' grades.

Integrity: Only authorized users should be allowed to modify data. For example, students may be allowed to see their grades, yet not allowed (obviously!) to modify them.

Availability: Authorized users should not be denied access. For example, an instructor who wishes to change a grade should be allowed to do so.

Authentication is the act of confirming the truth of an attribute of a datum or entity. This might involve confirming the identity of a person or software program.

2. Explain the terms security policy and security mechanism and how they are related. (Section 21.1)

Answer

Security policy is developed to describe what security measures must be enforced. In particular, we must determine what part of the data is to be protected and which users get access to which portions of the data.

Security mechanisms is the mechanisms which be utilized to enforce the Security policy, which also is the relation of security policy and security mechanism.

3. What is the main idea behind discretionary access control? What is the idea behind mandatory access control? What are the relative merits of these two approaches? (Section 21.2)

Answer

Discretionary access control is based on the concept of access rights, or privileges, and mechanisms for giving users such privileges. A privilege allows a user to access some data object in a certain manner (e.g., to read or to modify).

A user who creates a database object such as a table or a view automatically gets all applicable privileges on that object. The DBMS subsequently keeps track of how these privileges are granted to other users, and possibly revoked, and ensures that at all times only users with the necessary privileges can access an object.

Discretionary access control mechanisms, while generally effective, have certain weaknesses. In particular, a devious unauthorized user can trick an authorized user into disclosing sensitive data. **Mandatory access control** is based on systemwide policies that cannot be changed by individual users. In this approach each database object is assigned a security class, each user is assigned clearance for a security class, and rules are imposed on reading and writing of database objects by users. The DBMS determines whether a given user can read or write a given object based on certain rules that involve the security level of the object and the clearance of the user. These rules seek to ensure that sensitive data can never be 'passed on' to a user without the necessary clearance.

4. Describe the privileges recognized in SQL. In particular, describe privileges regarding SELECT, INSERT, UPDATE, DELETE and REFERENCES. For each privilege, indicate who acquires it automatically on a given table. (Section 21.3)

Answer:
Several privileges can be specified, including these:

- **SELECT:** The right to access (read) all columns of the table specified as the object, including columns added later through **ALTER TABLE** commands.
- **INSERT(column-name):** The right to insert rows with (non-null or nondefault) values in the named column of the table named as object. If this right is to be granted with respect to all columns, including columns that might be added later, we can simply use INSERT. The privileges **UPDATE(column-name)** and **UPDATE** are similar.
- **DELETE:** The right to delete rows from the table named as object.
- **REFERENCES(column-name):** The right to delete foreign keys (in other tables) that refer to the specified column of the table object. REFERENCES without a column name specified denotes this right with respect to all columns, including any that are added later.

A user who creates a base table automatically has all applicable privileges on it, along with the right to grant these privileges to other users. If a user has a privilege with the grant option, he or she can pass it to another user (with or without the grant option) by using the GRANT command.

5. How are the owners of privileges identified? In particular, discuss authorization ID and roles. (Section 21.3)

Answer:
Privileges are assigned in SQL-92 to authorization ids, which can denote a single user or a group of users; a user must specify an authorization id and, in many systems, a corresponding password before the DBMS accepts any commands from him or her.
Privileges are assigned to users (authorization ids, to be precise) in SQL-92. In the real world, privileges are often associated with a user's job or role within the organization. Many DBMSs have long supported the concept of a role and allowed privileges to be assigned to roles. Roles can then be granted to users and other roles. (Of courses, privileges can also be granted directly to users.)

6. What is an authorization graph? Explain SQL's GRANT and REVOKE commands in terms of their effect on this graph. In particular, discuss what happens when users pass on privileges that they receive from someone else. (Section 21.3)

Answer:
The effect of a series of GRANT commands can be described in terms of an authorization graph in which the nodes are users-technically, they are authorization ids-and the arcs indicate how privileges are passed. There is an arc from (the node for) user 1 to user 2 if user 1 executed a GRANT command giving a privilege to user 2; the arc is labeled with the descriptor for the GRANT command. A GRANT command has no effect if the same privileges have already been granted to the same grantee by the same grantor.

7. Discuss the difference between having a privilege on a table and on a view defined over the table. In particular, how can a user have a privilege (say. SELECT) over a view without also having it on all underlying tables? Who must have appropriate privileges on all underlying tables of the view? (Section 21.3.1)

Answer:
If a user has a privilege with the grant option, he or she can pass it to another user (with or without the grant option) by using the GRANT command. A user who creates a base table automatically has all applicable privileges on it, along with the right to grant these privileges to other users. A user who creates a view has precisely those privileges on the view that he or she has on every one of the view or base tables used to define the view. The user creating the view must have the SELECT privilege on each underlying table, of course, and so is always granted the SELECT privilege on the view. The creator of the view has the SELECT privilege with the grant option only if he or she has the SELECT privilege with the grant option on every underlying table. In addition, if the view is updatable and the user holds INSERT, DELETE, or UPDATE privileges (with or without the grant option) on the (single) underlying table, the user automatically gets the same privileges on the view.

8. What are objects, subjects, security classes, and clearances in mandatory access control? Discuss the Behl-LaPadula restrictions in terms of these concepts. Specifically, define the simple security property and the *-Property. (Section 21.4)

Answer:
The popular model for mandatory access control, called the Bell-LaPadula model, is described in terms of **objects** (e.g., tables, views, rows, and columns), **subjects** (e.g., users, programs), **security classes**, and **clearances**. Each database object is assigned a **security class**, and each subject is assigned **clearance** for a **security class**; we will denote the class of an object or subject A as class (A). The security classes in a system are organized according to a partial order, with a most secure class and a least secure class. For simplicity, we will assume that there are four classes: top secret (TS), secret (S),

confidential (C), and unclassified (U). In this system, TS > S > C > U, where A > B means that class A data is more sensitive than class B data.

The Bell-LaPadula model imposes two restrictions on all reads and writes of database objects:

1. **Simple Security Property:** Subject S is allowed to read object O only if class(S) ≥class (O). For example, a user with TS clearance can read a table with C clearance, but a user with C clearance is not allowed to read a table with TS classification.
2. ***-Property**: Subject S is allowed to write object O only if class(S)≤ class (O). For example, a user with S clearance can only write objects with S or TS classification.

9. What is a Trojan horse attack and how can it compromise discretionary access control? Explain how mandatory access control protects against Trojan horse attacks. (Section 21.4)

Answer:

A Trojan horse attack can be designed to accomplish any number of goals, but typically the intent is either pecuniary gain or spreading mayhem. For example, upon bringing the infected file onto your hard drive, the Trojan horse program might locate and send your bank information to the developer. The only limit to what a Trojan horse attack can accomplish is dependent on the limits of the developer's imagination and talent.

Example:

Suppose that student Tricky Dick wants to break into the grade tables of instructor Trustin Justin. Dick does the following:

- He creates a new table called MineAllMine and gives INSERT privileges on this table to Justin (who is blissfully unaware of all this attention, of course).
- He modifies the code of some DBMS application that Justin uses often to do a couple of additional things: first, read the Grades table, and next, write the result into MineAllMine.

Then he sits back and waits for the grades to be copied into MineAllMine and later undoes the modifications to the application to ensure that Justin does not somehow find out later that he has been cheated. Thus, despite the DBMS enforcing all discretionary access controls-only Justin's authorized code was allowed to access Grades-sensitive data is disclosed to an intruder.

Mandatory access control mechanisms are aimed at addressing such loopholes in discretionary access control. The popular model for mandatory access control, called the Bell-LaPadula model. Let us consider how such a mandatory control mechanism might have foiled Tricky Dick. The Grades table could be classified as S, Justin could be given clearance for S, and Tricky Dick could be given a lower clearance (C). Dick can only create objects of C or lower classification; thus, the table MineAllMine can have at most the classification C. When the application program running on behalf of Justin (and therefore with clearance S) tries to copy Grades into MineAllMine, it is not allowed to do so because class (MineAllMine) < class (application), and the *-Property is violated.

10. What do the terms multilevel table and polyinstantiation mean? Explain their relationship and how they arise in the context of mandatory access control. (Section 21.4.1)

Answer:
To apply mandatory access control policies in a relational DBMS, a security class must be assigned to each database object. The objects can be at the granularity of tables, rows, or even individual column values. Let us assume that each row is assigned a security class. This situation leads to the concept of a **multilevel table**, which is a table with the surprising property that users with different security clearances will see a different collection of rows when they access the same table.
The presence of data objects that appear to have different values to users with different clearances is called **polyinstantiation**.

11. What are covert channels and how can they arise when both discretionary and mandatory access controls are in place? (Section 21.4.2)

Answer:
Even if a DBMS enforces the mandatory access control scheme, information can flow from a higher classification level to a lower classification level through indirect means, called covert channels.

12. Discuss the DoD security levels for database systems. (Section 21.4.2)

Answer:
The DoD requirements can be described in terms of security levels A, B, C, and D of which A is the most secure and D is the least secure.
Level C requires support for discretionary access control. It is divided into sublevels C1 and C2; C2 also requires some degree of accountability through procedures such as login verification and audit trails. Level B requires support for mandatory access control. It is subdivided into levels B1, B2, and B3. Level B2 additionally requires the identification and elimination of covert channels. Level B3 additionally requires maintenance of audit trails and the designation of a security administrator (usually, but not necessarily, the DBA). Level A, the most secure level, requires a mathematical proof that the security mechanism enforces the security policy!

13. Explain why a simple password mechanism is insufficient for authentication of users who access a database remotely, say, over the Internet. (Section 21.5)

Answer:
When a DBMS is accessed from a secure location, we can rely upon a simple password mechanism for authenticating users. However, suppose our friend Sam wants to place an order for a book over the Internet. This presents some unique challenges: Sam is not

even a known user (unless he is a repeat customer). From Amazon’s point of view, we have an individual asking for a book and offering to pay with a credit card registered to Sam, but is this individual really Sam? From Sam's point of view, he sees a form asking for credit card information, but is this indeed a part of Amazon's site, and not a rogue application designed to trick him to revealing his credit card number?
This example illustrates the need for a more sophisticated approach to authentication than a simple password mechanism.

14. What is the difference between symmetric and public-key encryption? Give examples of will-known encryption algorithms of both encryption .What is the main weakness of symmetric encryption and how is this addressed in public-key encryption? (Section 21.5.1)

Answer:
In symmetric encryption, the encryption key is also used as the decryption key. The ANSI Data Encryption Standard (DES), which has been in use since 1977, is a well-known example of symmetric encryption. It uses an encryption algorithm that consists of character substitutions and permutations. The main weakness of symmetric encryption is that all authorized users must be told the key, increasing the likelihood of its becoming known to an intruder.
In Public-key encryption, each authorized user has a public encryption key, known to everyone, and a private decryption key, known only to him or her. The encryption schema proposed by Hjvest, Sharnir, and Adlernan, called RSA, is a well-known example of public-key encryption. Since the private decryption keys are known only to their owners, the weakness of DES is avoided.

15. What is the choice of encryption and decryption keys in public-key encryption and how they are used to encryption and decryption data? Explain the role of one-way function. (Section 21.5.1)

Answer:
A central issue for public-key encryption is ho\v encryption and decryption keys are chosen. Technically, public-key encryption algorithms rely on the existence of one-way functions, whose inverses are complicatedly very hard to determine. The RSA algorithm, for example, is based on the observation that, although checking whether a given number is a prime is easy, determining the prime factors of a nonprime number is extremely hard.

16. What are certification authorities and why are they needed? Explain how certificates are issued to sites and validated by a browser using the SSL protocol; discuss the role of the session key. (Section 21.5.2)

Answer:

In cryptography, a certificate authority or certification authority (CA) is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others (relying parties) to rely upon signatures or assertions made by the private key that corresponds to the public key that is certified. In this model of trust relationships, a CA is a trusted third party that is trusted by both the subject (owner) of the certificate and the party relying upon the certificate. CAs are characteristic of many public key infrastructure (PKI) schemes.

A number of companies serve as certification authorities, e.g., Verisign. Amazon generates a public encryption key e_A(and a private decryption key) and sends the public key to Verisign. Verisign then issues a certificate to Amazon that contains the following information:

<*Verisign Amazon, http:// www. amazon. com,* e_A >

The certificate is encrypted using Verisign's own private key, which is known to (i.e., stored in) Internet Explorer, Netscape Navigator, and other browsers.

When Sam comes to the Amazon site and wants to place an order, his browser, running the SSL protocol, asks the server for the Verisign certificate. The browser then validates the certificate by decrypting it (using Verisign's public key) and checking that the result is a certificate issued by Verisign, and that the URL it contains is that of the server it is talking to. (Note that an attempt to forge a certificate will fail because certificates are encrypted using Verisign's private key, which is known only to Verisign.) Next, the browser generates a random session key, encrypt it using Amazon's public key (which it obtained from the validated certificate and therefore trusts), and sends it to the Amazon server.

From this point on, the Amazon server and the browser can use the session key (which both know and are confident that only they know) and a symmetrical encryption protocol like AES or DES to exchange securely encrypted messages. Messages are encrypted by the sender and decrypted by the receiver using the same session key. The encrypted messages travel over the Internet and may be intercepted, but they cannot be decrypted without the session key. It is useful to consider why we need a session key; after all, the browser could simply have encrypted Sam's original request using Amazon's public key and sent it securely to the Amazon server. The reason is that, without the session key, the Amazon server has no way to securely send information back to the browser. A further advantage of session keys is that symmetric encryption is computationally much faster than public key encryption. The session key is discarded at the end of the session.

17. If a user connects to a site using the SSL protocol, explain why there is still a need to login. Explain the use of SSL to protect passwords and other sensitive information being exchanged. What is the secure electronic transaction protocol? What is the added value over SSL? (Section 21.5.2)

Answer:
Because Amazon has no assurance that the user running the browser is actually the person claimed to be, and not someone who has stolen his credit card, even a user connects to a site using the SSL protocol.
When Sam comes to the Amazon site and wants to place an order, his browser, running the SSL protocol, asks the server for the Verisign certificate. The browser then validates the certificate by decrypting it (using Verisign's public key) and checking that the result is a certificate issued by Verisign, and that the URL it contains is that of the server it is talking to. (Note that an attempt to forge a certificate will fail because certificates are encrypted using Verisign's private key, which is known only to Verisign.) Next, the browser generates a random session key, encrypt it using Amazon's public key (which it obtained from the validated certificate and therefore trusts), and sends it to the Amazon server.
From this point on, the Amazon server and the browser can use the session key (which both know and are confident that only they know) and a symmetrical encryption protocol like AES or DES to exchange securely encrypted messages. Messages are encrypted by the sender and decrypted by the receiver using the same session key. The encrypted messages travel over the Internet and may be intercepted, but they cannot be decrypted without the session key.
Secure Electronic Transaction (SET) was a communications protocol standard for securing credit card transactions over insecure networks, specifically, the Internet. SET was not itself a payment system, but rather a set of security protocols and formats that enabled users to employ the existing credit card payment infrastructure on an open network in a secure fashion.

18. A digital signature facilitates secure exchange of messages. Explain what it is and how it goes beyond simply encrypting messages. Discuss the use of message signatures to reduce the cost of encryption. (Section 21.5.3)

Answer:
A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity).
For example, If Elmer encrypts messages for Betsy using her public key, and vice versa, they can exchange information securely but cannot authenticate the sender. Someone who wishes to impersonate Betsy could use her public key to send a message to Elmer, pretending to be Betsy.

A clever use of the encryption scheme, however, allows Elmer to verify whether the message was indeed sent by Betsy. Betsy encrypts the message using her private key and then encrypts the result using Elmer's public key. When Elmer receives such a message, he first decrypts it using his private key and then decrypts the result using Betsy's public key. This step yields the original unencrypted message. Furthermore, Elmer can be certain that the message was composed and encrypted by Betsy because a forger could not have known her private key, and without it the final result would have been nonsensical, rather than a legible message. Further, because even Elmer does not know Betsy's private key, Betsy cannot claim that Elmer forged the message.

If authenticating the sender is the objective and hiding the message is not important, we can reduce the cost of encryption by using a message signature. A signature is obtained by applying a one-way function (e.g., a hashing scheme) to the message and is considerably smaller. We encode the signature as in the basic digital signature approach, and send the encoded signature together with the full, unencrypted message. The recipient can verify the sender of the signature as just described, and validate the message itself by applying the one-way function and comparing the result with the signature.

19. What is the role of the database administrator with respect to security? (Section 21.6.1)

The database administrator (DBA) plays an important role in enforcing the security related aspects of a database design. In conjunction with the owners of the data, the DBA will probably also contribute to developing a security policy. The DBA has a special account, which we will call the system account, and is responsible for the overall security of the system. In particular the DBA deals with the following:

- **Creating new accounts:** Each new user or group of users must be assigned an authorization id and a password. Note that application programs that access the database have the same authorization id as the user executing the program.
- **Mandatory control issues:** If the DBMS supports mandatory control-some customized systems for applications with very high security requirements (for example, military data) provide such support-the DBA must assign security classes to each database object and assign security clearances to each authorization id in accordance with the chosen security policy.

The DBA is also responsible for maintaining the audit trail, which is essentially the log of updates with the authorization id (of the user who is executing the transaction) added to each log entry. This log is just a minor extension of the log mechanism used to recover from crashes. Additionally, the DBA may choose to maintain a log of all actions, including reads, performed by a user. Analyzing such histories of how the DBMS was accessed can help prevent security violations by identifying suspicious patterns before an intruder finally succeeds in breaking in, or it can help track down an intruder after a violation has been detected.

20. Discuss the additional security loopholes introduced in statistical databases. (Section 21.6.2)

Answer:
A statistical database is one that contains specific information on individuals or events but is intended to permit only statistical queries. For example, if we maintained a statistical database of information about sailors, we would allow statistical queries about average ratings, maximum age, and so on, but would not want to allow queries about individual sailors. Security in such databases poses some new problems because it is possible to infer protected information (such as an individual sailor's rating) from answers to permitted statistical queries. Such inference opportunities represent covert channels that can compromise the security policy of the database.

Advantages & Disadvantages of XML

Advantages of XML

- It is a simultaneously human- and machine-readable format.
- It supports Unicode, allowing almost any information in any written human language to be communicated.
- It can represent the most general computer science data structures: records, lists and trees.
- Its self-documenting format describes structure and field names as well as specific values.
- The strict syntax and parsing requirements make the necessary parsing algorithms extremely simple, efficient, and consistent.
- XML is heavily used as a format for document storage and processing, both online and offline.
- It is based on international standards.
- It allows validation using schema languages such as XSD and Schematron, which makes effective unit-testing, firewalls, acceptance testing, contractual specification and software construction easier.
- The hierarchical structure is suitable for most (but not all) types of documents.
- It manifests as plain text files, which are less restrictive than other proprietary document formats.
- It is platform-independent, thus relatively immune to changes in technology.
- Forward and backward compatibility are relatively easy to maintain despite changes in DTD or Schema.
- Its predecessor, SGML, has been in use since 1986, so there is extensive experience and software available.

Disadvantages of XML

- XML syntax is redundant or large relative to binary representations of similar data.
- The redundancy may affect application efficiency through higher storage, transmission and processing costs.
- XML syntax is too verbose relative to other alternative ‘text-based’ data transmission formats.
- No intrinsic data type support: XML provides no specific notion of “integer”, “string”, “boolean”, “date”, and so on.
- The hierarchical model for representation is limited in comparison to the relational model or an object oriented graph.
- Expressing overlapping (non-hierarchical) node relationships requires extra effort.
- XML namespaces are problematic to use and namespace support can be difficult to correctly implement in an XML parser.
- XML is commonly depicted as “self-documenting” but this depiction ignores critical ambiguities.