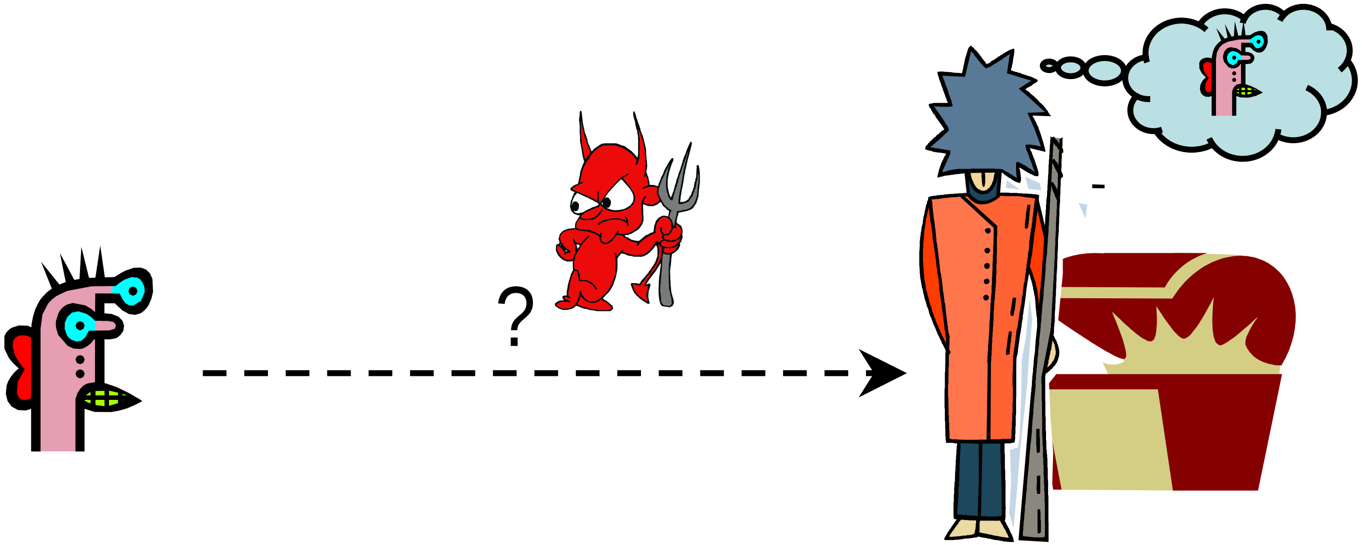
Network Security

– Authentication Techniques

Basic Problem



How do you prove to someone that you are who you claim to be?

Any system with access control must solve this problem

Authentication

- Authentication can be defined as determining an identity to the required level of assurance
- Authentication is the first step in any cryptographic solution
  - Because unless we know who is communicating, there is no point in encryption what is being communicated

Authentication

- Authentication is any process by which a system verifies the identity of a user who wishes to access it

- Authentication may be implemented using Credentials, each of which is composed of a User ID and Password. Alternately, Authentication may be implemented with Smart Cards, an Authentication Server or even a Public Key Infrastructure

Many Ways to Prove Who You Are

- What you know
  - Passwords/Secret key
- Where you are
  - IP address
- What you are
  - Biometrics (e.g. fingerprint)
- What you have
  - Secure tokens/smart card/ ATM card

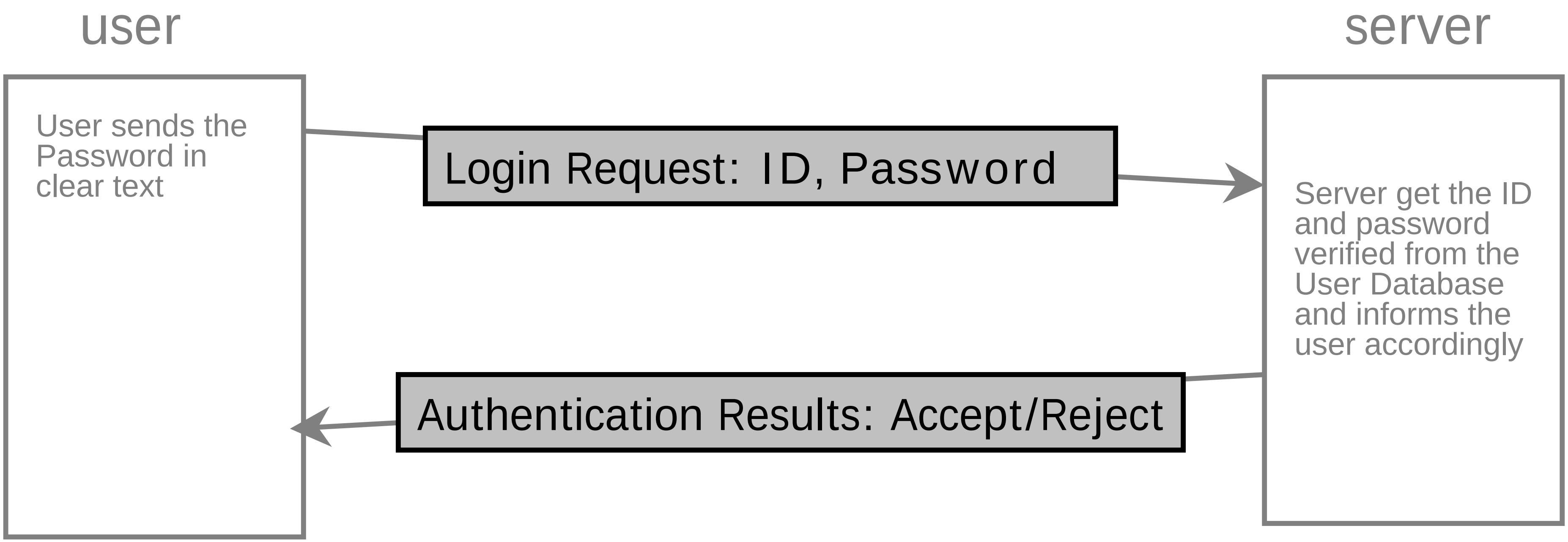
Passwords

- A password is a string of alphabets, numbers and special characters, which is supposed to be known only to the entity (usually person) that is being authenticated
- Password Based Authentication
  - **Clear Text Passwords**

    Simplest Password based Authentication Mechanism

    - How it works?
      - Prompt for user ID and Password
      - User enters user ID and Password
      - User ID and Password Validation
      - Authentication Result
      - Inform user accordingly

Password Based Authentication



user

server

User sends the Password in clear text

Login Request: ID, Password

Server get the ID and password verified from the User Database and informs the user accordingly

Authentication Results: Accept/Reject

User Authentication using Clear Text Password

- Clear Text Passwords are being sent from Client to Server.
- User Database stores Passwords in Clear Text Format

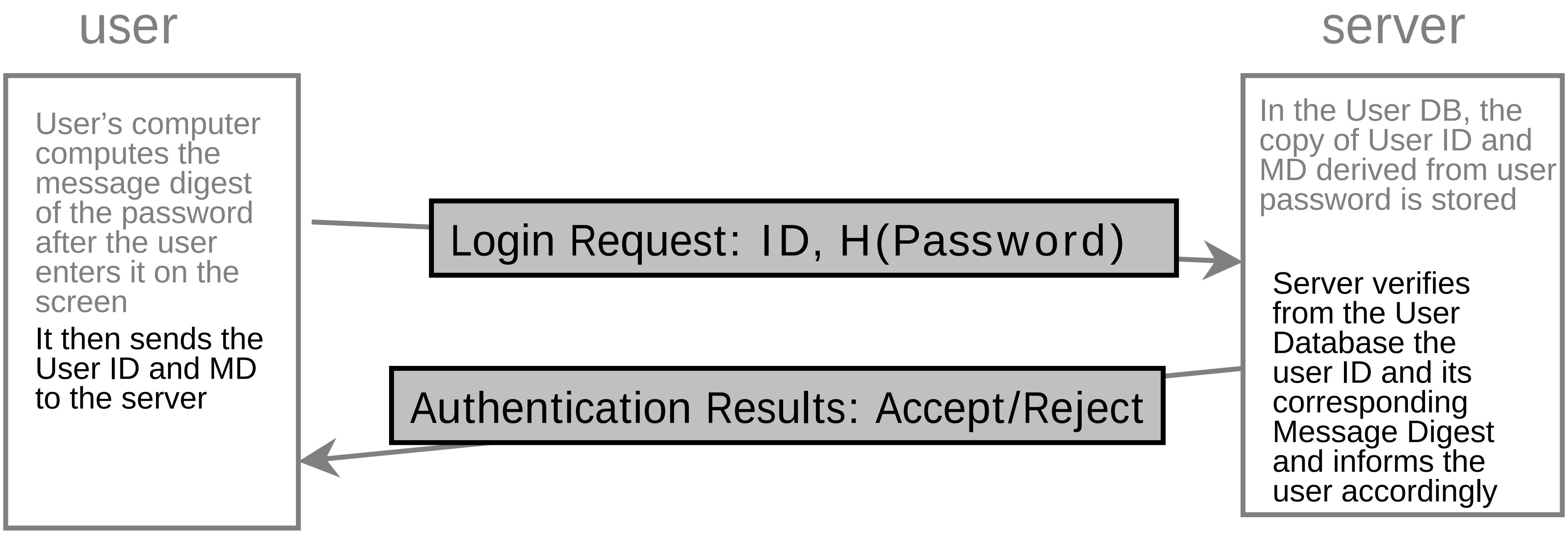
Passwords Based Authentication

- Problems with Clear Text Passwords
  - Database contains Passwords in clear text
    - It is advised that password should not be stored in clear text in databases
    - Instead the passwords should be stored in encrypted form in database
  - Password travel in clear text from user's computer to the server
    - If the attacker breaks into the communication link, he can easily obtain the clear text password

Password Based Authentication

- Something Derived from Password
  - **Message Digests of the Passwords**
    - Storing Message Digests as derived passwords in the user database
    - User Authentication
      - When a user need to be authenticated, the user enters the ID and Password.
      - User's computer computes the message digest of the password
      - User's computer sends the user ID and computed Message Digest to the server for authentication

Password Based Authentication



user

server

User's computer computes the message digest of the password after the user enters it on the screen

It then sends the User ID and MD to the server

Login Request: ID, H(Password)

Authentication Results: Accept/Reject

In the User DB, the copy of User ID and MD derived from user password is stored

Server verifies from the User Database the user ID and its corresponding Message Digest and informs the user accordingly

- H(Password) = Message Digest Derived from the User Password

User Authentication Involving Message Digests of the Password

Passwords Based Authentication

- Problems with the Message Digests of the Passwords
  - An Attacker cannot compute the original password back from the message digest of the password
  - But he can simply copy the User ID and the corresponding Message Digest of the password, and submit them after some time to the same server as a part of the new login request.
  - The server has no way of knowing that this login attempt is not from a legitimate user, but actually an attacker.
  - This is called as **REPLAY ATTACK**, because the attacker simply replays the sequence of the actions of a normal user

Passwords Based Authentication

- **Adding Randomness**
  - To improve the security and to detect a replay attack we need to add a bit of unpredictability or randomness to the earlier schemes
  - This will ensure that the replay attack is foiled
  - **Steps**

1. Storing Message Digests as derived passwords in the user database
   - User Ids and corresponding MDs are stored in user Database with the server
2. User sends a login request
   - Containing only user ID

Passwords Based Authentication

- **Adding Randomness … (cont)**
  - Steps…

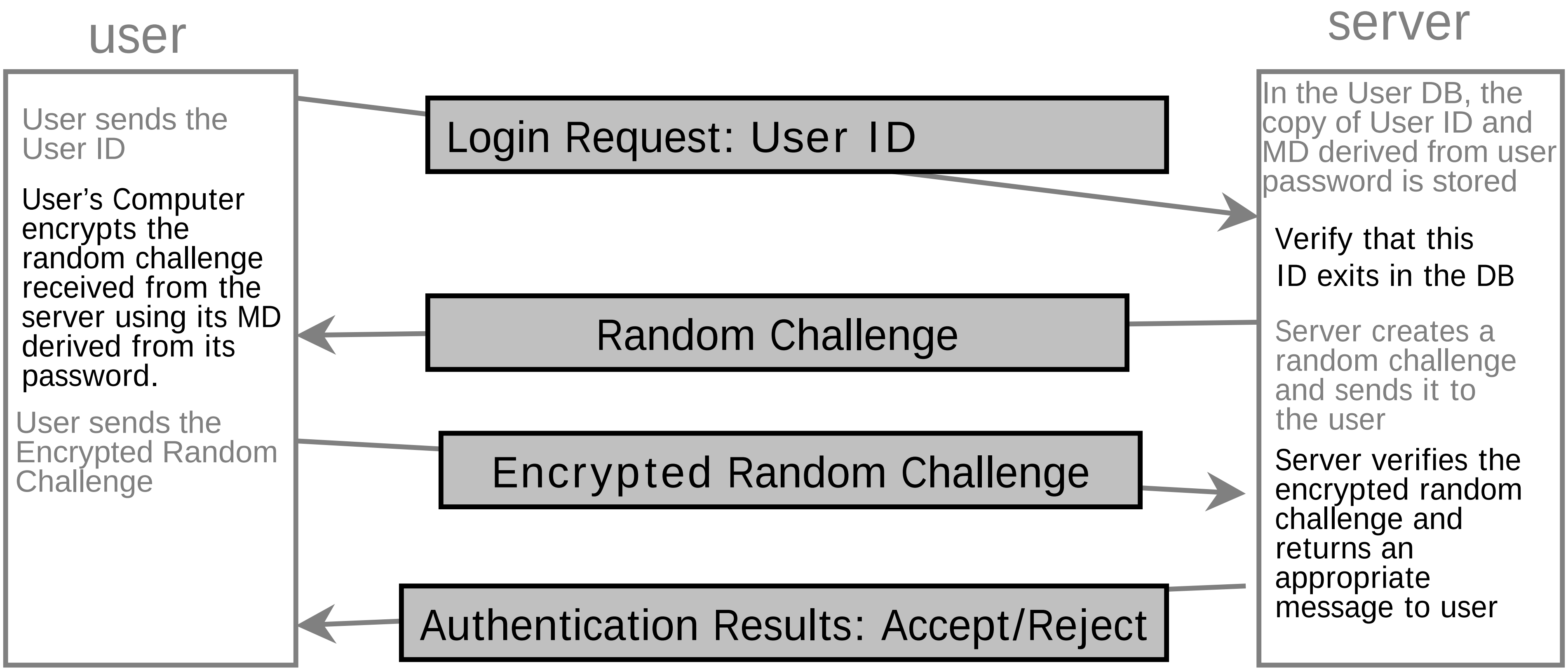
    3. Server Creates a random Challenge
       - Server first verifies the validity of user ID
       - Then it sends a random challenge (a random number) to the user
       - Random challenge travels as plain text from server to user computer
    4. User Signs the Random Challenge with the Message Digest of the Password
       - User Computer’s computes the Message Digest (MD) of its password
       - User Computer’s Encrypts the Random Challenge by using MD of the Password. (symmetric key encryption)
       - User Computer’s sends the random challenge, which is encrypted with the message digest of the password to the server .

Passwords Based Authentication

- **Adding Randomness … (cont)**
  - Steps…
    5. Server Verifies the Encrypted Random Challenge from the user
       - Server can do the verification in two ways
       - Either decrypting the Random Challenge and comparing the challenge values.
       - Or it can encrypt the Random Challenge by the MD of the password and compare the two encrypted entities
    6. Server returns an appropriate message back to the user

Note: *The Random Challenge value is different every time. Therefore the random challenge encrypted with the MD of password would also be different. Therefore replay attacks can easy be detected*

Password Based Authentication



user

server

User sends the User ID

User's Computer encrypts the random challenge received from the server using its MD derived from its password.

User sends the Encrypted Random Challenge

Login Request: User ID

Random Challenge

Encrypted Random Challenge

Authentication Results: Accept/Reject

In the User DB, the copy of User ID and MD derived from user password is stored

Verify that this ID exits in the DB

Server creates a random challenge and sends it to the user

Server verifies the encrypted random challenge and returns an appropriate message to user

Adding Randomness in Password Based Authentication

Problems with the Passwords

- Typically an organization has a number of applications, networks, shared resources an intranets
  - These applications may have varying needs of security measures, each resource may demands its own user name and password
  - In that case end users/network administrators have to keep a large number of user ids and passwords to be used with different applications
- Password Maintenance is a very big concern for system administrations
- Organizations specify Password Policies
- There exit other authentication mechanisms as well besides Password based authentication

Authentication Tokens

- It is an extremely useful alternative to a password
- These small devices are usually of the size of a small key chain
- Usually an authentication Token has the following features
  - Processor
  - LCD for displaying outputs
  - Battery
  - Optionally a small keypad for entering information
  - Optionally a real-time clock
- Each Authentication Token is pre-programmed with a unique number called as a random seed or just **seed**
- **The seed value forms the basis for ensuring the uniqueness of the output produced by the token**

Authentication Token

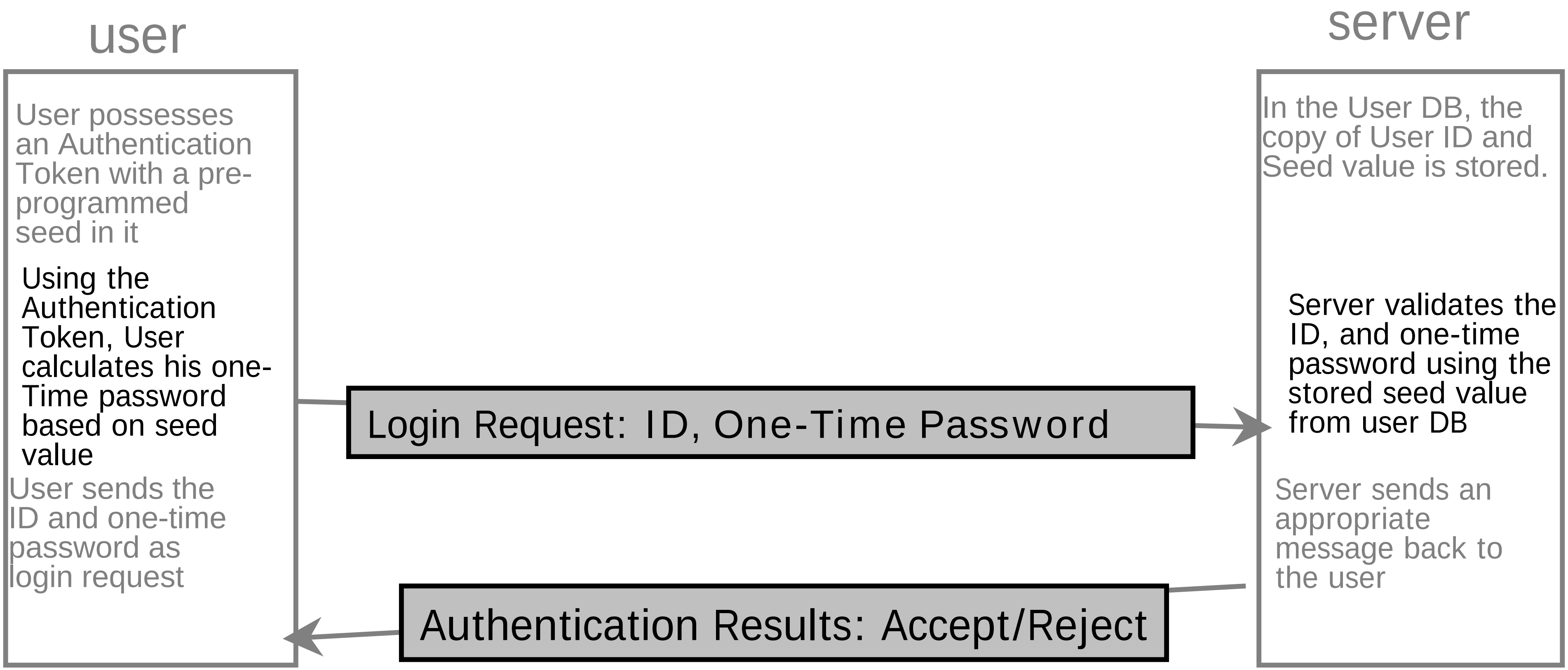
- Step Involved in Authentication Token

1. Creation of a Token
   - Created by the Authentication servers that are designed to use with authentication tokens
   - A unique value i.e. a seed is automatically placed or pre-programmed inside each token by the server
   - Server also keeps a copy of the seed against the user ID in the user database
   - Seed can be conceptually considered as a user password
   - Difference is that the user password is known to the user, seed value remains unknown to the user
2. Use of the Token
   - An Authentication Token automatically generates pseudorandom numbers called **one-time passwords**.
   - One time passwords are generated randomly by authentication tokens using seed value

Authentication Token … (cont)

- – When a user wants to be authenticated by any server, the user will get a screen to enter user ID and the latest one-time password
- – The users enters its ID and gets is latest one-time password from the authentication token
- – The user ID and password travels to the server as a part of the login request
- – Server verifies using some mechanism that this one-time password is created using the valid seed value

Authentication Token



user

User possesses an Authentication Token with a pre-programmed seed in it

Using the Authentication Token, User calculates his one-Time password based on seed value

User sends the ID and one-time password as login request

Login Request: ID, One-Time Password

server

In the User DB, the copy of User ID and Seed value is stored.

Server validates the ID, and one-time password using the stored seed value from user DB

Server sends an appropriate message back to the user

Authentication Results: Accept/Reject

Authentication Tokens

Multifactor Authentication

- What if the Authentication Token device gets stolen
  - PIN numbers are used to generate a one-time passwords with the authentication token devices
- Multifactor Authentication
  - What you know
    - Passwords/Secret key
  - What you are
    - Biometrics (e.g. fingerprint)
  - What you have
    - Secure tokens/smart card/ ATM card

Authentication Tokens

- Password is a 1-factor authentication
  - It is something you know
- Authentication Token are **2-factor** authentication
  - You must have something
    - The authentication token itself
  - You must know something
    - PIN to protect it

Certificate Based Authentication

- This is based on the Digital Certificates of the user
- In PKI, the digital certificates are used for secure digital transactions.
- The digital certificates in PKI can also be re-used for user authentication as well
- This is a stronger mechanism as compared to password based authentication
  - Issue
    - Misuse of someone else's certificate
    - To tackle such issues, certificate based authentication is also made 2 factor process (have something and know something)

Certificate Based Authentication

- How does Certificate Based Authentication works?

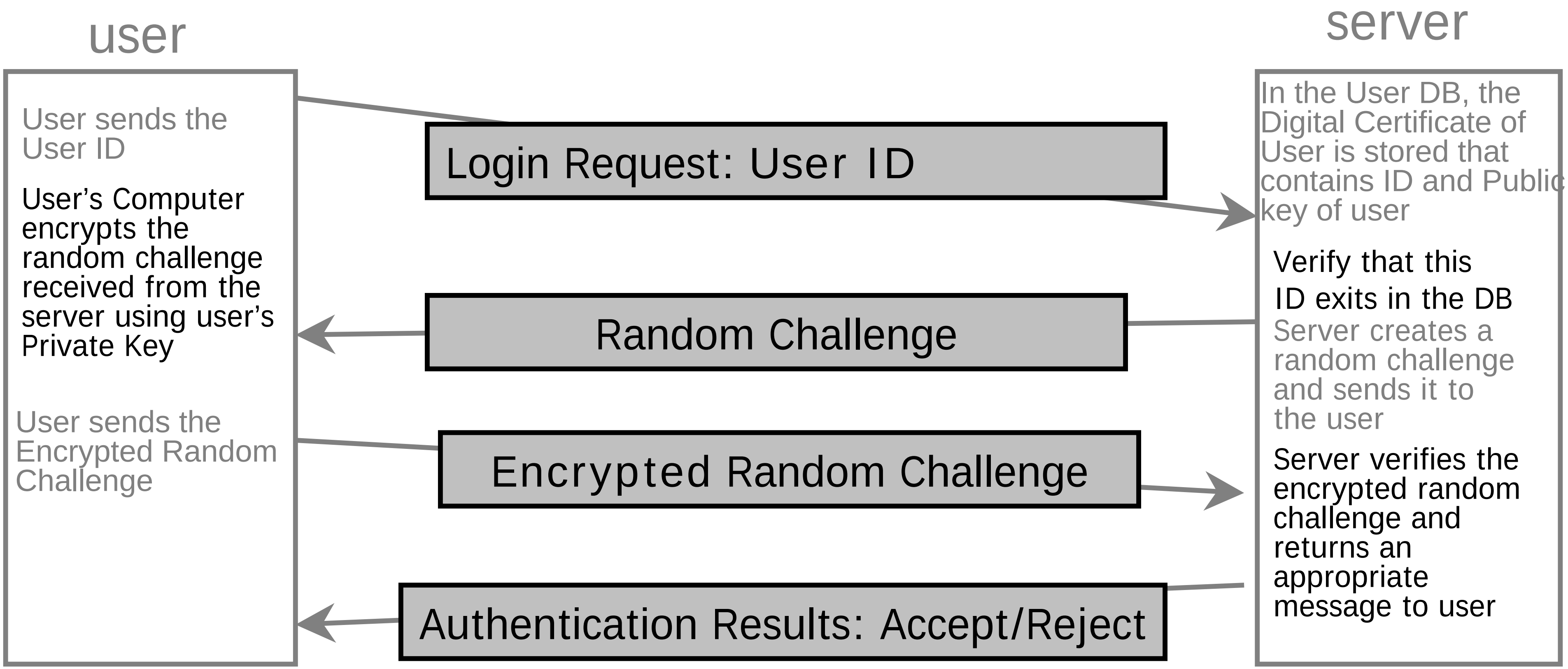
1. Creation, Storage and Distribution of Digital Certificates
   - Certificates are created by CA, sent to user as well as a copy to the server, where we have to implement certificate based authentication
2. Login Request
   - User sends its ID only
3. Server Creates a Random Challenge
   - User ID validity is checked
   - Sends random challenge in plain text to user

Certificate Based Authentication

- How does Certificate Based Authentication works? **… (Cont)**

  4. User Signs the Random Challenge
     - User signs the random challenge received from Server by using its Private Key
     - User's private key is stored in a file in user computer
     - To access its private key file, user has to give a correct password
     - User sends the signed random challenge to the server
  5. Server returns an appropriate message back to the user

Certificate Based Authentication



user

User sends the User ID

User's Computer encrypts the random challenge received from the server using user's Private Key

User sends the Encrypted Random Challenge

server

In the User DB, the Digital Certificate of User is stored that contains ID and Public key of user

Verify that this ID exits in the DB

Server creates a random challenge and sends it to the user

Server verifies the encrypted random challenge and returns an appropriate message to user

Login Request: User ID

Random Challenge

Encrypted Random Challenge

Authentication Results: Accept/Reject

Certificate Based Authentication

Use of Smart Cards

- The use of Smart Cards is related to Certificate Based Authentication
- This is because the smart cards allows the generation of public-private key pairs within the card
- They also support the storage of digital certificates within the card
- The private key always remain in the smart card in a secure fashion
- The public key and the certificate is exposed outside
- Also the smart cards are capable of performing cryptographic functions such as encryption, decryption, message digest creation and signing within the card
  - Thus during the certificate based authentication, the signing of random challenge sent by the server can be performed inside the card

Smart Cards

- Portable
  - Have its own pros and cons
- Must be used to perform selective cryptographic mechanisms
  - Such approaches must be used to first generate the message digest of large document say 10MB of size by some software (inside the computer), and then use the smart card to digitally sign the created message digest

Problems and issues in Smart Cards

- Lack of standardization and inter-operability between smart cards vendors
- Smart card reader are not yet a part of a desktop computer like hard disk drive or floppy drives
- Non-availability of smart card reader driver software
- Non-availability of smart card aware cryptographic service software
- Cost of smart cards and card reader is high

Biometric Authentication

- A biometric device works on the basis of some human characteristics, such as fingerprints, voice or the pattern of lines in the iris of your eye
- The user database contains a sample of user's biometric characteristics
- During the authentication, the user is required to provide another sample of the users' biometric characteristic.
- This is matched with the one in the database, and if the two samples are same, the user is considered to be a valid one.
- The samples produced during every authentication process can vary slightly. (e.g. cuts on the finger)
- An approximate match can be acceptable

Biometric Authentication

- Any Biometric Authentication System defines two configurable parameters:
  - False Accept Ratio (FAR)
    - It is a measurement of the chance that a user who should be rejected is actually accepted by a system as good enough
  - False Reject Ratio (FRR)
    - It is a measurement of the chance that a user who should be accepted as valid is actually rejected by a system as not good enough

- Thus FAR and FRR are exactly opposite to each other

Best Authentication Solution

- The best authentication solution may be considered as a combination of the password/PIN, a smart card and biometrics.
- It covers all the three key aspects related to authentication
  - Who you are,
  - What you have,
  - What you know
    - However this can turn out to be an extremely complex system to build

References

- Network Security A Beginner's Guide
  - By Eric Maiwald
- Network Security first-step
  - By Tom Thomas
- Designing Network Security
  - By Cisco Press
- Cryptography and Network Security
  - 3rd Edition, by William Stallings
- Mastering Network Secuity
  - By Chris Brenton and Cameron Hunt
- Network Security
  - By Atul Kahate