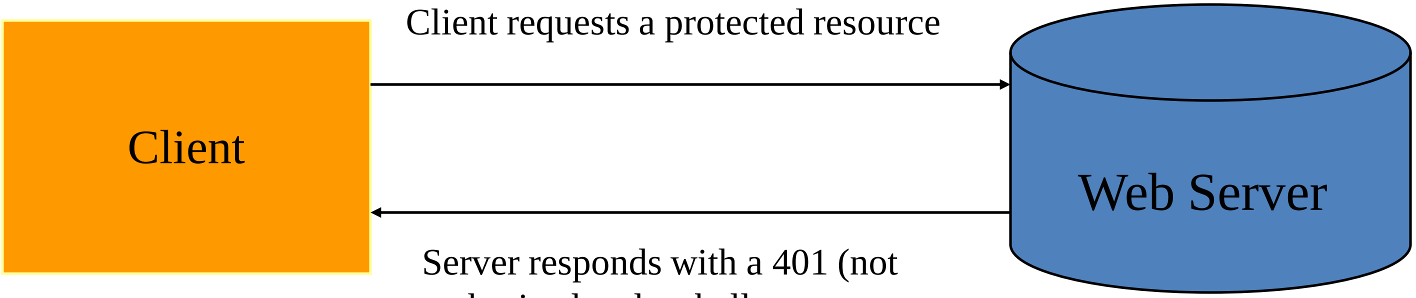
Web Security

HTTP Authentication

- Protect web content from those who don’t have a “need to know”
- Require users to authenticate using a userid/password before they are allowed access to certain URLs
- HTTP/1.1 requires that when a user makes a request for a protected resource the server responds with a authentication request header
  - WWW-Authenticate
    - contains enough pertinent information to carry out a “challenge-response” session between the user and the server



Server responds with a 401 (not authorized and a challenge request for the client to authenticate

Client Response

- Well established clients like Firefox, Internet Explorer …. will respond to the challenge request (WWW-Authenticate) by presenting the user with a small pop-up window with data entry fields for
  - userid
  - password
  - a Submit button and a Cancel button
- entering a valid userid and password will post the data to the server, the server will attempt authentication and if authenticated will serve the originally requested resource.

WWW-Authenticate

- The authentication request received by the browser will look something like:
  - WWW-Authenticate = Basic realm="defaultRealm"
    - Basic indicates the HTTP Basic authentication is requested
    - realm indicates the context of the login
      - realms hold all of the parts of security puzzle
        - Users
        - Groups
        - ACLs (Access Control Lists)
- Basic Authentication
  - userid and password are sent base 64 encoded (might as well be plain text)
  - hacker doesn't even need to unencode all he has to do is "replay" the blob of information he stole over and over (this is called a "replay attack")

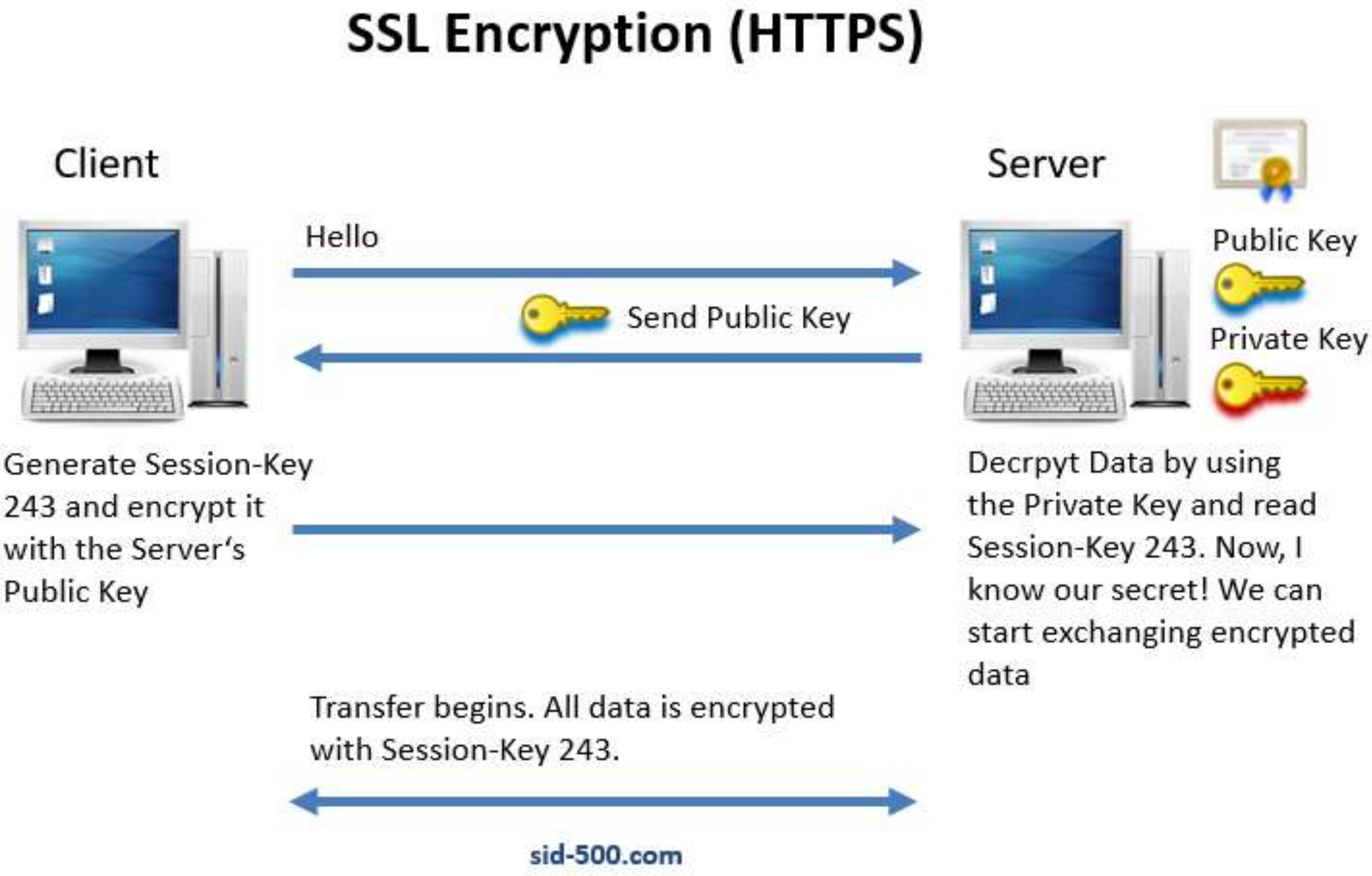
WWW-Authenticate

- Digest Authentication
  - attempts to overcome the shortcomings of Basic Authentication
  - WWW-Authenticate = Digest realm="defaultRealm" nonce="Server SpecificString"
  - see RFC 2069 for description of nonce, each nonce is different
  - the nonce is used in the browser in a 1-way function (MD5, SHA-1....) to encode the userid and password for the server, this function essentially makes the password good for only one time
- Common browsers don't use Digest Authentication but an applet could as an applet has access to all of the Java Encryption classes needed to create the creation of a Digest.

WWW-Authenticate

- Secure Sockets Layer (SSL)
  - Invented by Netscape and made public domain for everyone’s use
  - An additional layer to the TCP/IP stack that sits between the Application and Transport layers
    - ensures that all application data is encrypted but TCP/IP headers are not
    - usually run on port 443 (default HTTPS port)
- Public Key Cryptography
  - owner of a private key sends a public key to all who want to communicate with him (keys are both prime factors of a large (1024 bit) number). Owner keeps the private key secret and uses it to decrypt information sent to him that has been encrypted with the public-key
  - RSA algorithm is most notable public-key cipher algorithm
- Digital Certificates
  - issued by a disinterested third party (ex. Verisign)
  - the Certificate contains the public-key for the specific Web Server and a digital signature of the certifying authority

WWW-Authenticate



SSL Encryption (HTTPS)

Generate Session-Key 243 and encrypt it with the Server's Public Key

Decrpyt Data by using the Private Key and read Session-Key 243. Now, I know our secret! We can start exchanging encrypted data

Transfer begins. All data is encrypted with Session-Key 243.

sid-500.com

back to SSL

- Once a secure session is established the source requests the destinations certificate (sent in the http header (uncncrypted))
- once the source accepts the authenticity of the certificate it uses the public-key from the certificate to encrypt the generated session key for protecting the conversation between the source and destination.
- Session is encrypted using a symmetric cipher (slow)
- conversation is encrypted using an asymmetric cipher (fast)
- its done this way to speed up overall communications, strong encryption (slow) is used as little as possible while weaker encryption is used for most exchanges
- actual cipher algorithms are negotiated on a per-session basis