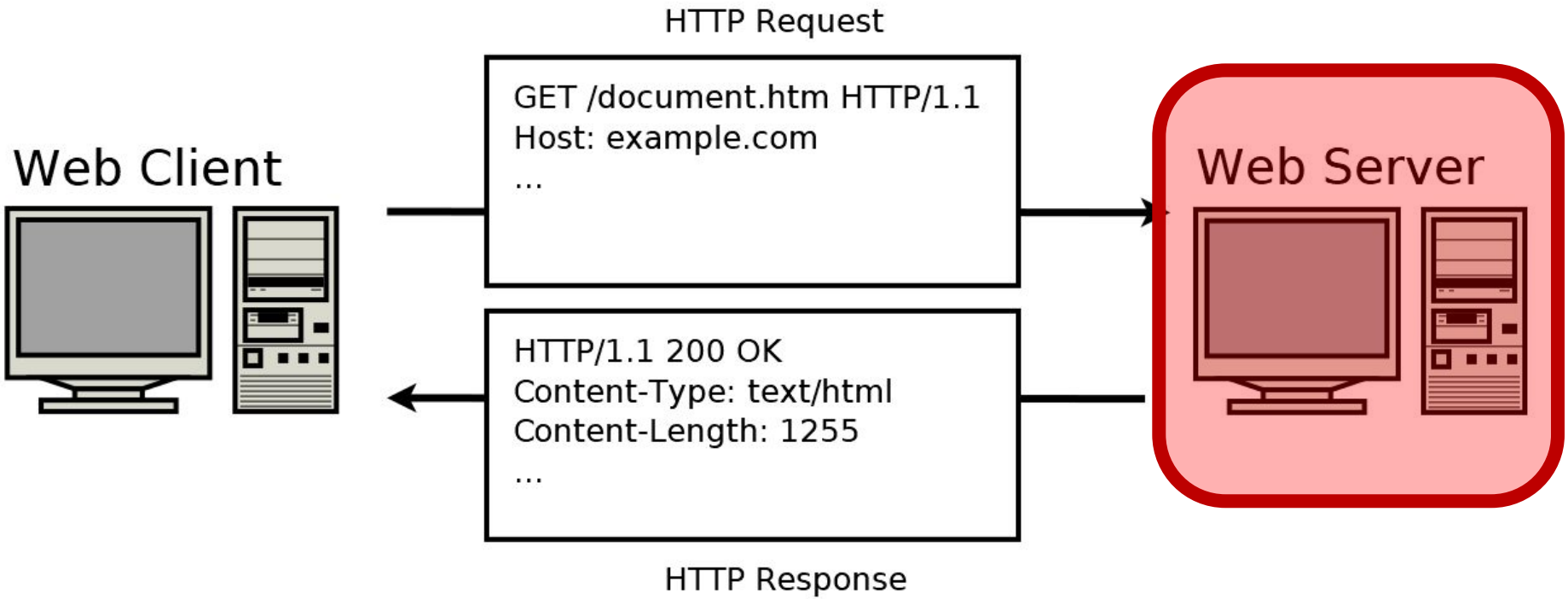
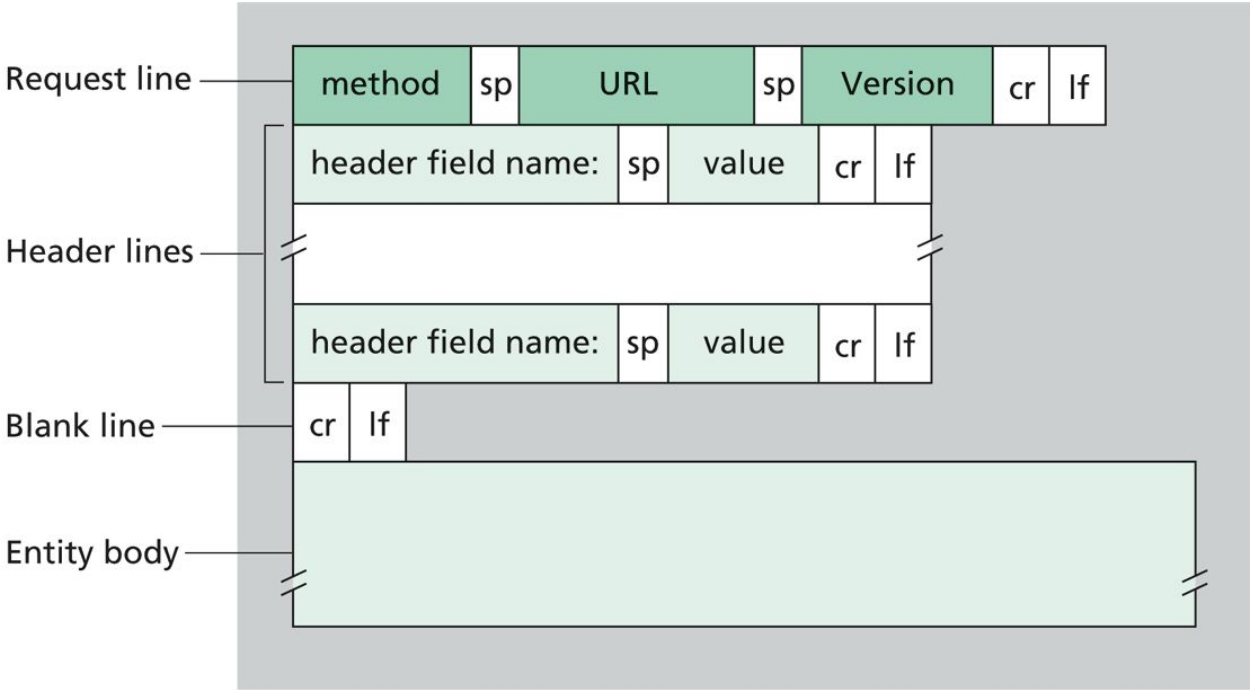
Recitation 9

Internet Technology (Section 01)

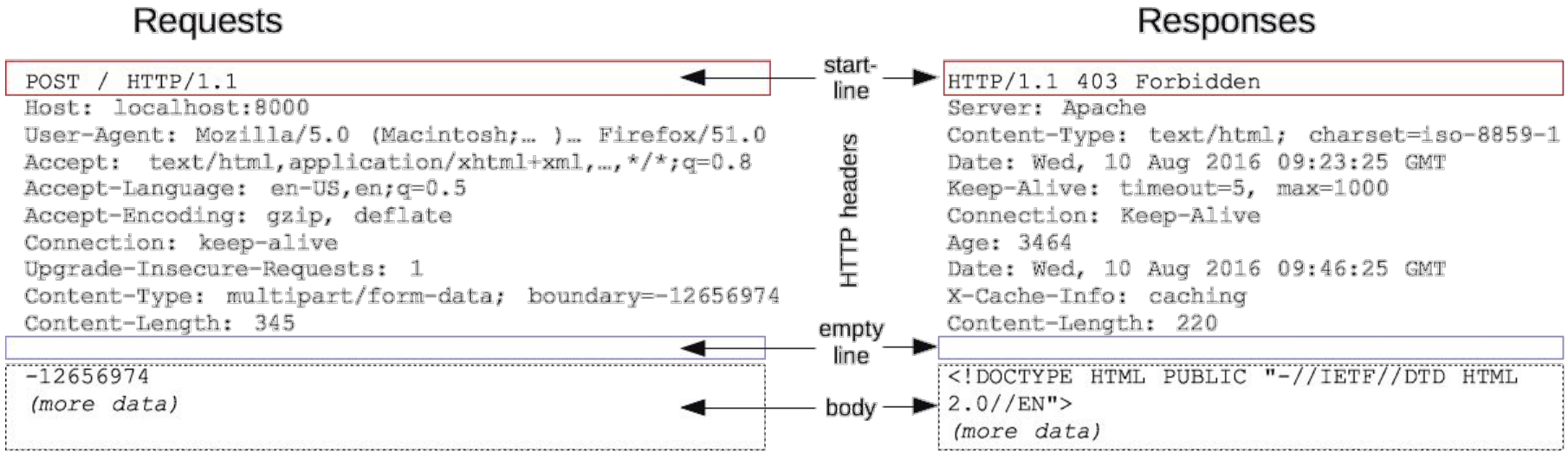
Project 3 – HTTP Server



HTTP Message Format



HTTP Message Example



HTTP Request Methods

Method	Description
GET	Request for resource from server
POST	Submit data to the server
HEAD	Same as GET but does not return the body
PUT	The data within the request must be stored at the URL supplied, replacing any existing data.
DELETE	Delete a resource
OPTIONS	Return the HTTP methods supported by the server
CONNECT	Client requests the HTTP proxy to forward a TCP connection to some destination. Used to create a TCP/IP tunnel for secure connections using HTTP proxies.

HTTP Response Status Codes

Success codes	200 OK
	201 Created
	202 Accepted
	203 Non-Authoritative information
	204 No Content
	205 Reset Content
	206 Partial Content
Client side errors	400 Bad Request
	401 Unauthorized
	402 Payment Required
	403 Forbidden
	404 Not Found
	405 Method Not Allowed
	409 Conflict
Informational	413 Payload Too Large
	415 Unsupported Media Type
	422 Unprocessable Entity
	429 Too Many Requests
	451 Unavailable For Legal Reasons
	500 Internal Server Error
	503 Service Unavailable
	504 Gateway Timeout
Redirectional	301 Moved Permanently
	302 Found (Moved temporarily)
	304 Not Modified

Parsing HTTP Messages

```
m = """GET /hello.htm HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Cookie: yummy_cookie=choco; tasty_cookie=strawberry
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
"""
#make newlines CRLF - not for a real HTTP message
m = str.replace(m, '\n', '\r\n')
lines = m.split("\r\n")
start_line = lines[0]
method,target,version = start_line.split(" ")
headers = {}
for header in lines[1:]:
    if header == "": break #reached body
    hkey,hval = header.split(": ",1)
    headers[hkey] = hval
print(method,target,version)
print(headers)
```

Cookies must be further parsed into another dictionary

Cookies

- HTTP is a stateless protocol
- Cookies are used to maintain a state
  - Mostly used to maintain state between browser and server
    - “Is this request from the same browser?”

Cookies

session-id used to keep user logged in…

Cookies

session-id used to keep user logged in…

Cookies

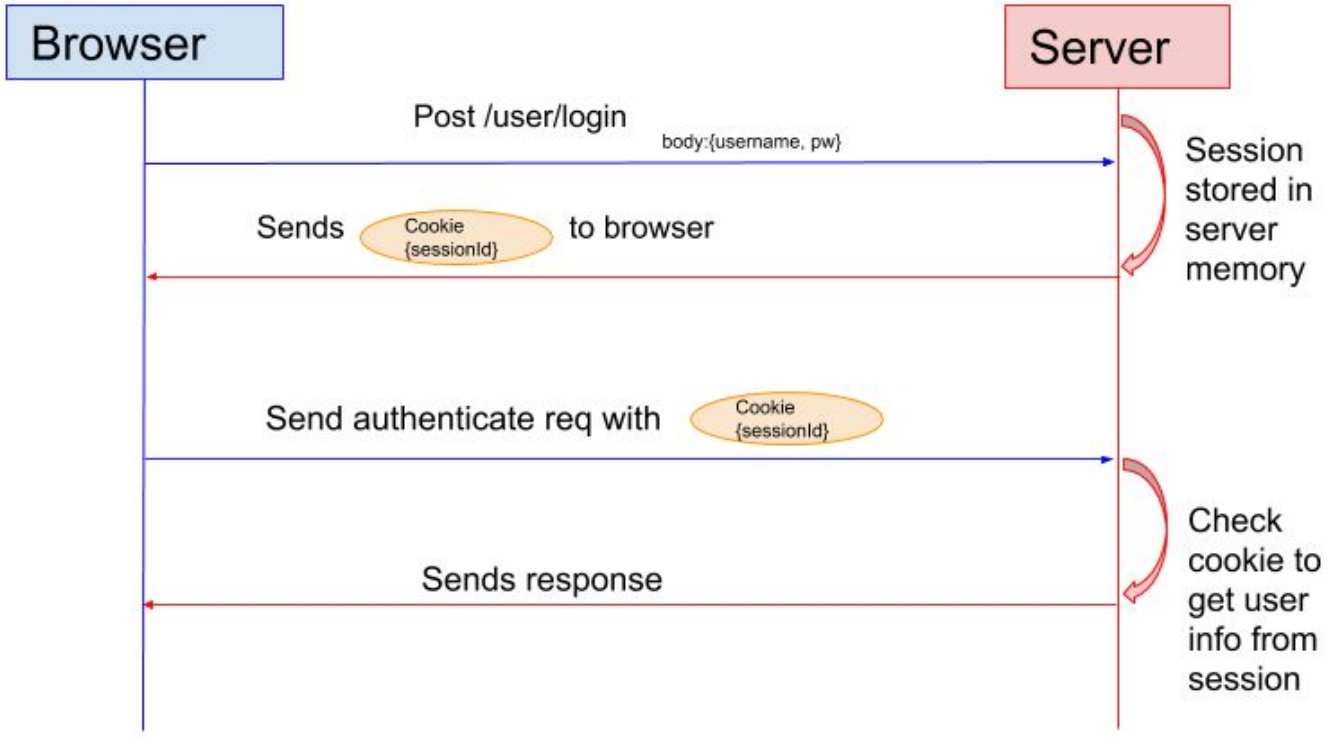
session-id used to keep user logged in…

Cookies

session-id should be a completely random value to prevent guessing

session-id used to keep user logged in…

Cookie-based User Authentication



Cookie Format

▼ Request Headers

:authority: www.geeksforgeeks.org

:method: GET

:path: /http-headers-cookie/

:scheme: https

accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3

accept-encoding: gzip, deflate, br

accept-language: en-US,en;q=0.9

cache-control: max-age=0

cookie: G_ENABLED_IDPS=google; _ga=GA1.2.236891924.1569526010; __gads=ID=f8deb276b85d6f74:T=1569559579:S=ALNI_Ma9kGxkZVOd23UT2B6UIi0-iHksPw; __utmz=245605906.1569597787.4.4.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=(not%20provided); __utma=245605906.236891924.1569526010.1569592113.1569597786.4; geeksforgeeks_consent_status=dismiss; gfguserName=Sabya_Samadder%2FeyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvd3d3LmdlZWtzZm9yZ2V1a3Mub3JnXC8iLCJpYXQiOjE1NzIzNDM3NjAsImV4cCI6MTU3NDkzNTc2MCwiaGFuZGxlIjoiU2FieWFfU2FtYWRkZXIiLCJ1dWlkIjoiYjA2NzA0Njc3MDMzMmY4Y2EyMDcxMDM4OWJjMWVkNjEifQ.f5Bky9sW46uX53XGJupbTHQPhSvjgr9_MCV5whZFzkBSCEZR_n4w-5Imj5eSa7TXuFS1r6NI1_VRnuz7Au0P_H-u6SEATsOVSkhssEMx0L6oj5NDQw1jk3ZAreK7dk_xyRLgnHsTJws40GbLi9__Yirrp9q2BNGztaMVTtXsqrW9knMAsOVKNmhEGM7bh9fpsEYh3PqKRRag4WI1dGRaZZ6Y-orBA91Srj9oyzqYOOFK3zmXd9pHKW7b_ffH5sheGW2EM7uwtjoMiGA7oc6RuG0G8sdpPPYL6Ktfkai2g_oHPRahoRsZ_UUQT3jNY91bHt75gxi2PMWw6KOUsNcujA; wordpress_test_cookie=WP+Cookie+check; AKA_A2=A

referer: https://www.google.com/

sec-fetch-mode: navigate

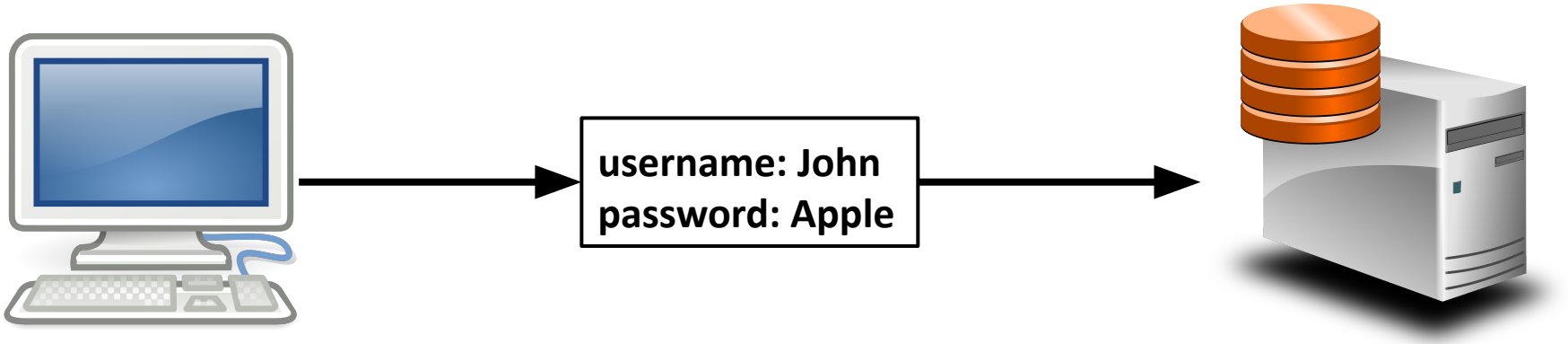
sec-fetch-site: cross-site

sec-fetch-user: ?1

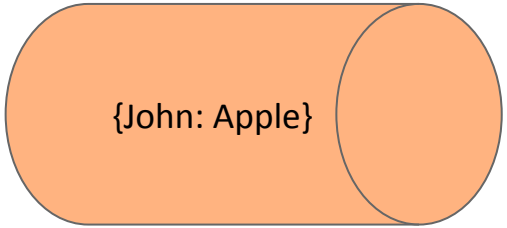
upgrade-insecure-requests: 1

user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36

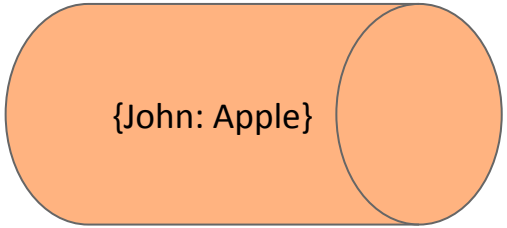
Hashing Passwords



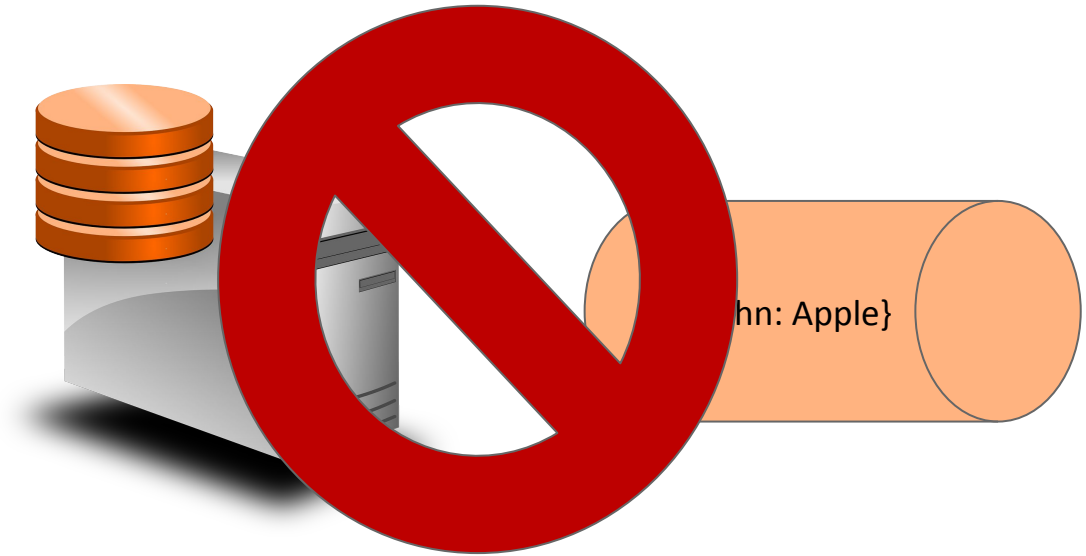
Hashing Passwords



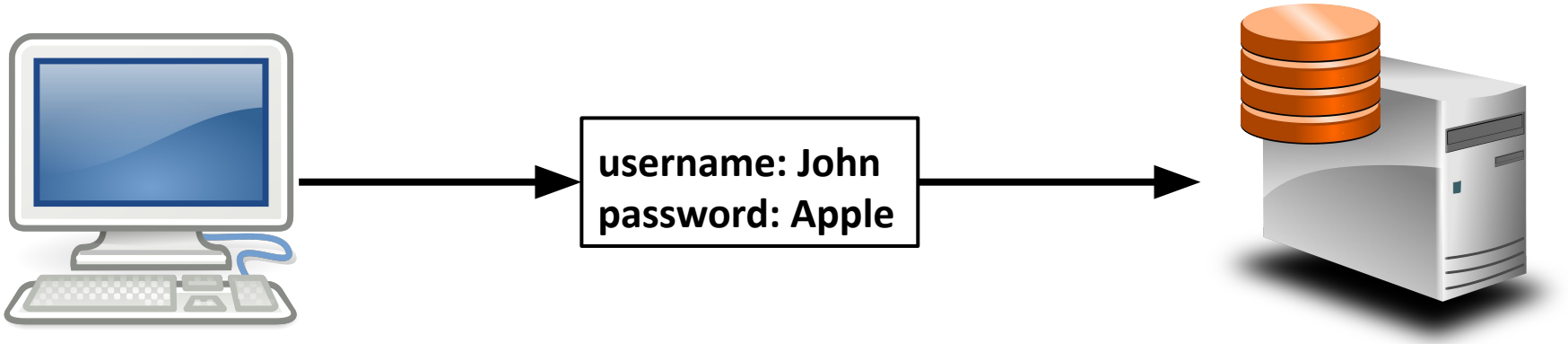
Hashing Passwords



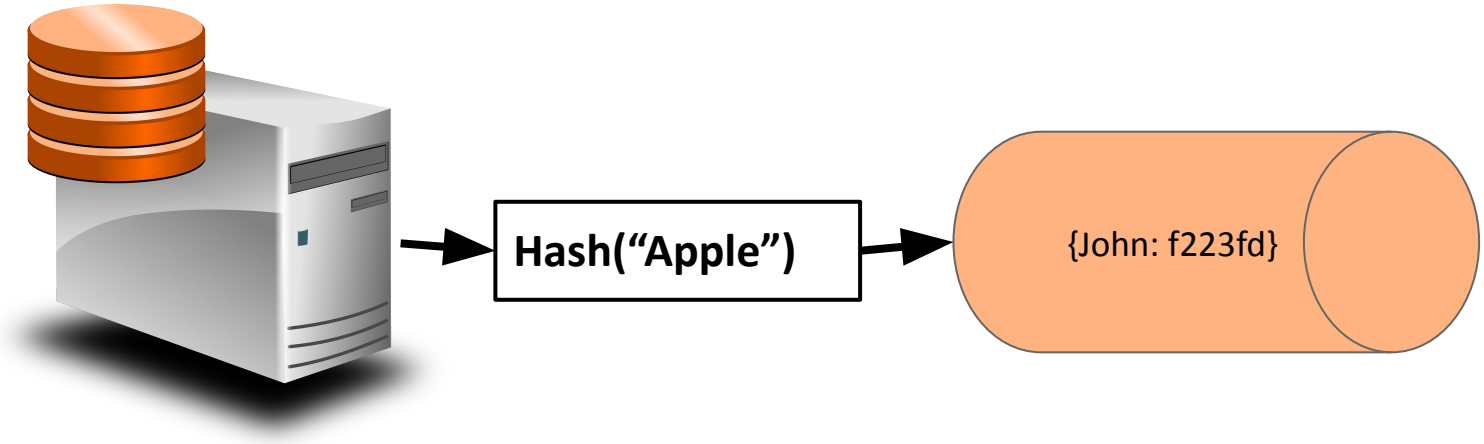
Hashing Passwords



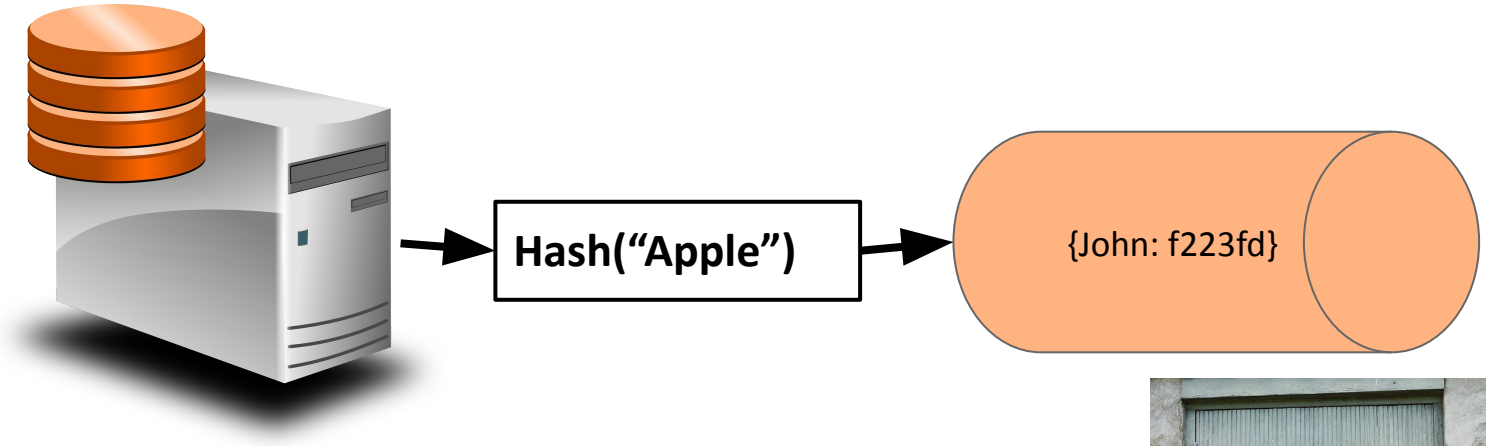
Hashing Passwords



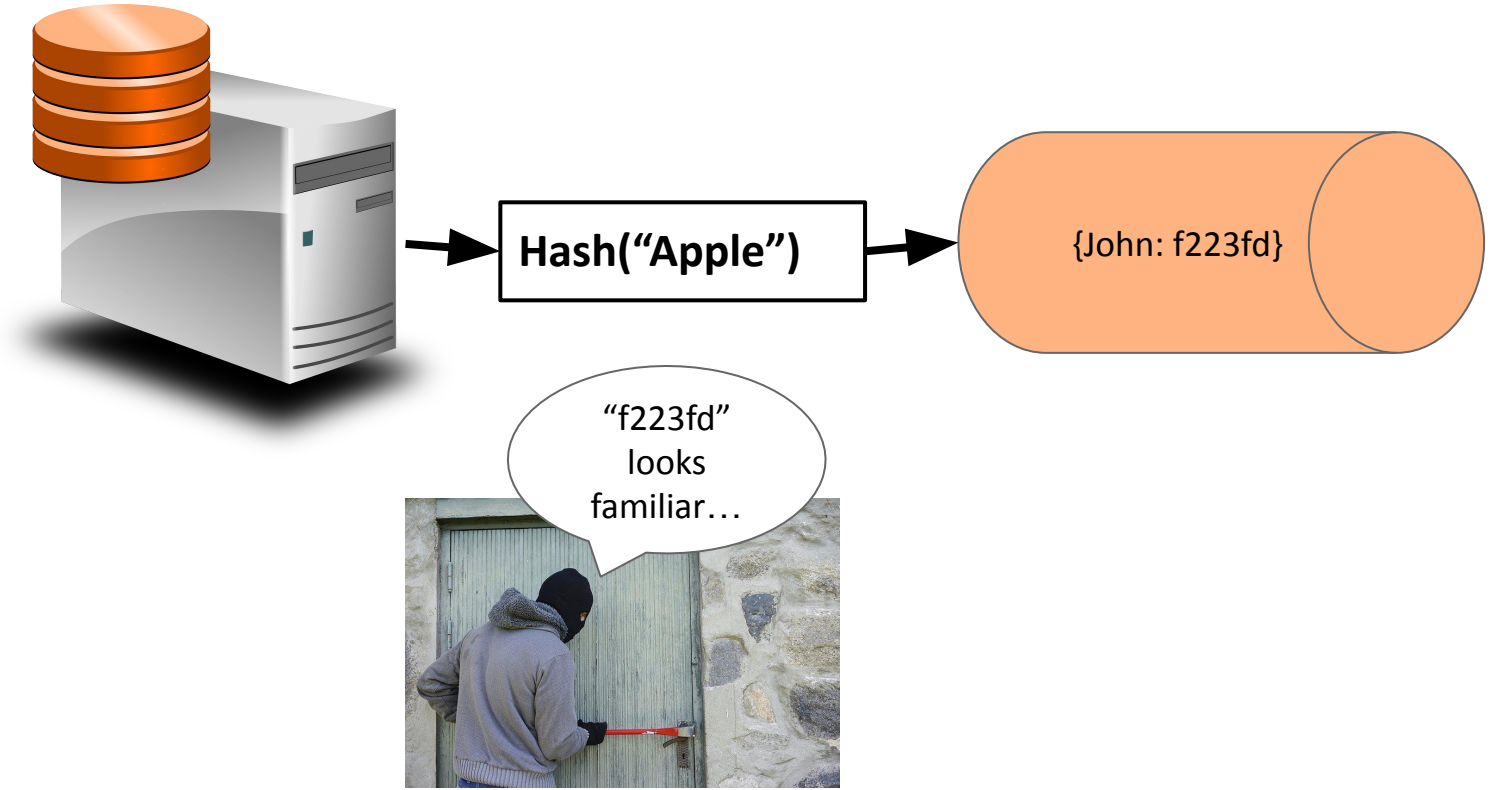
Hashing Passwords



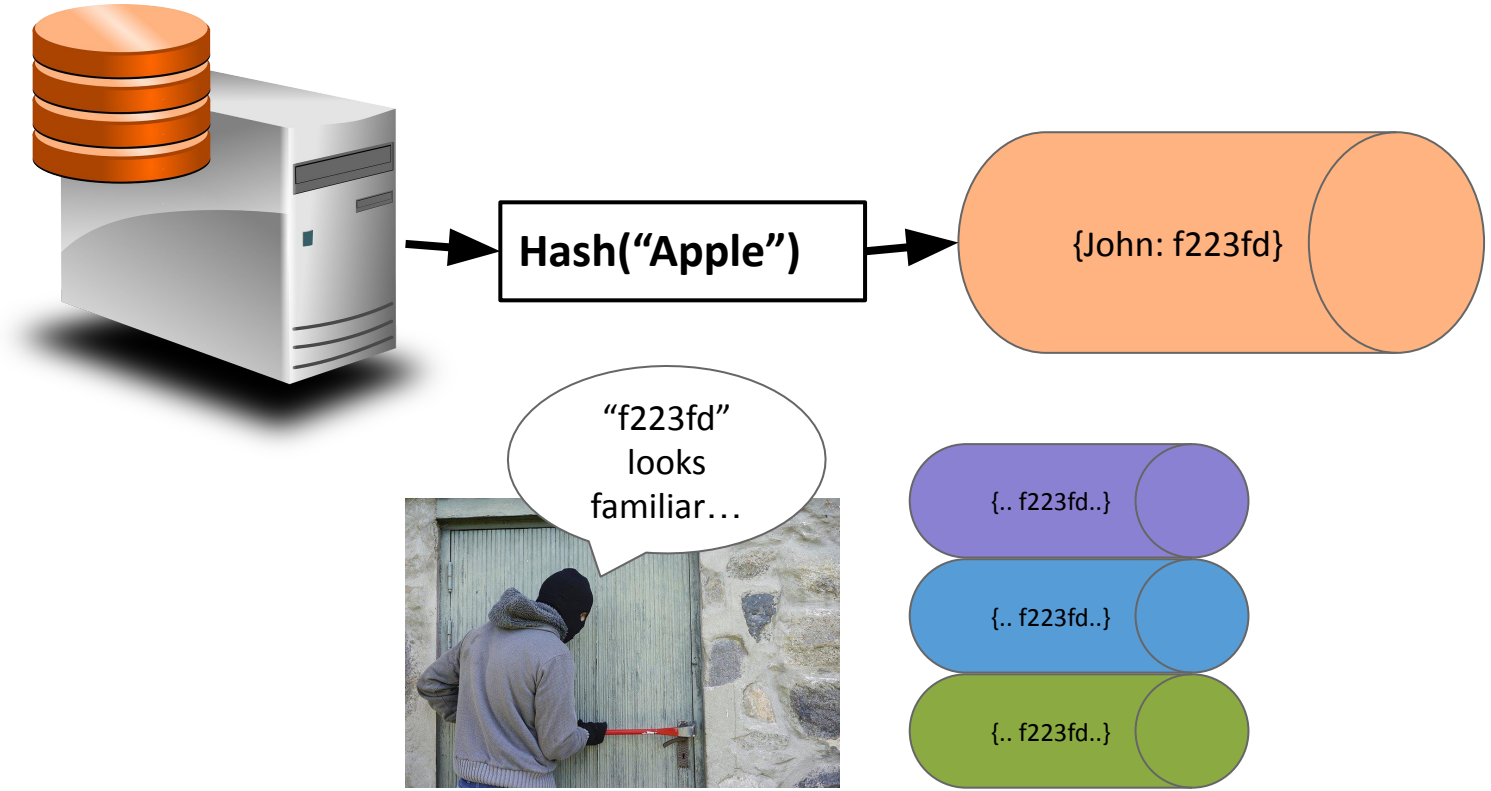
Hashing Passwords



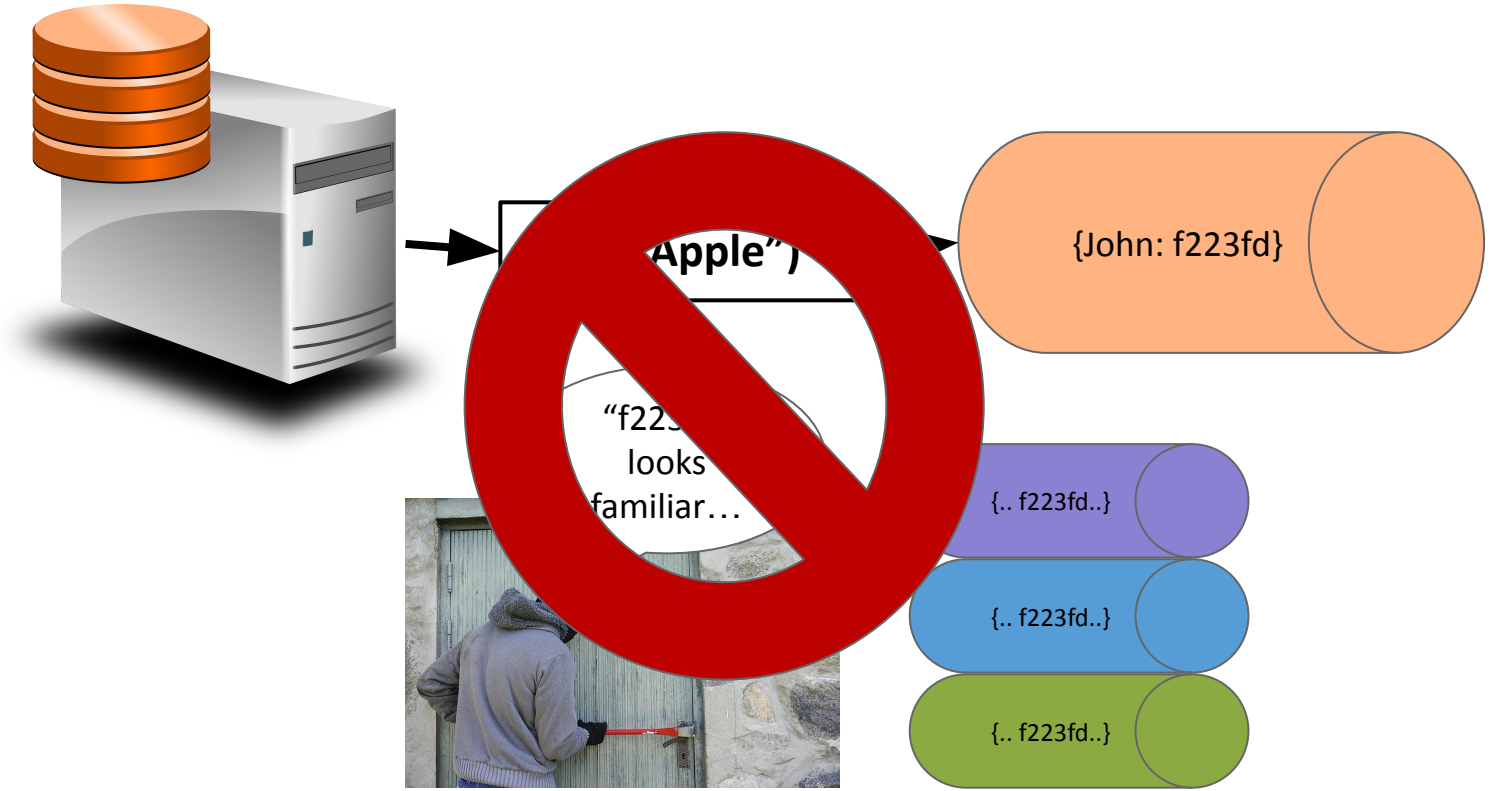
Hashing Passwords



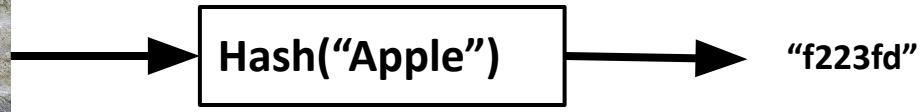
Hashing Passwords



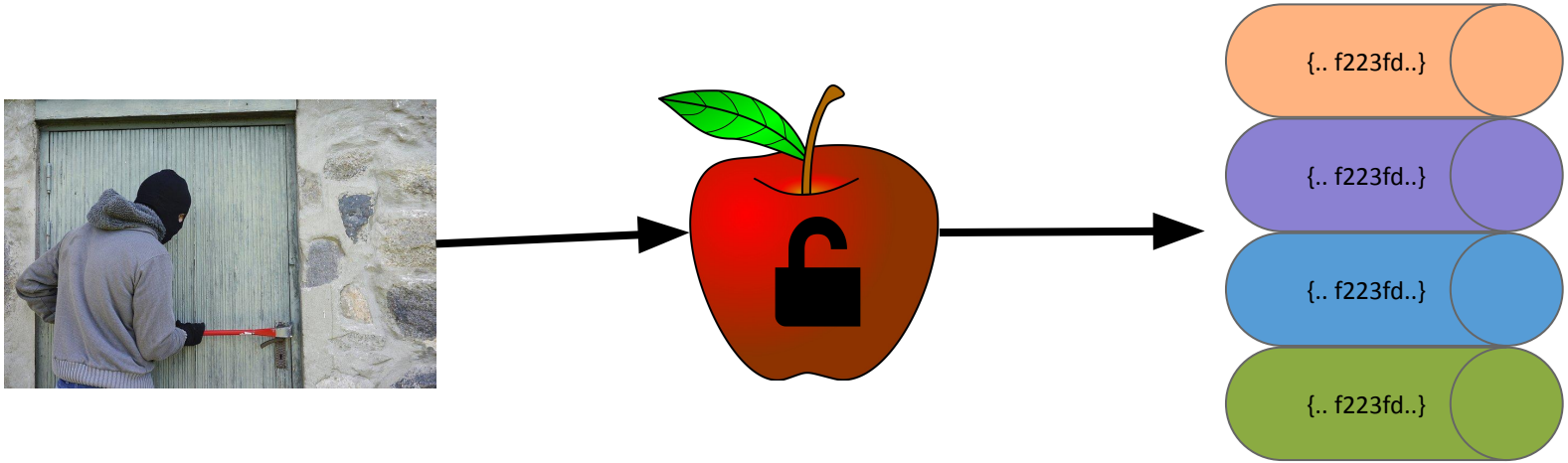
Hashing Passwords



Hashing Passwords



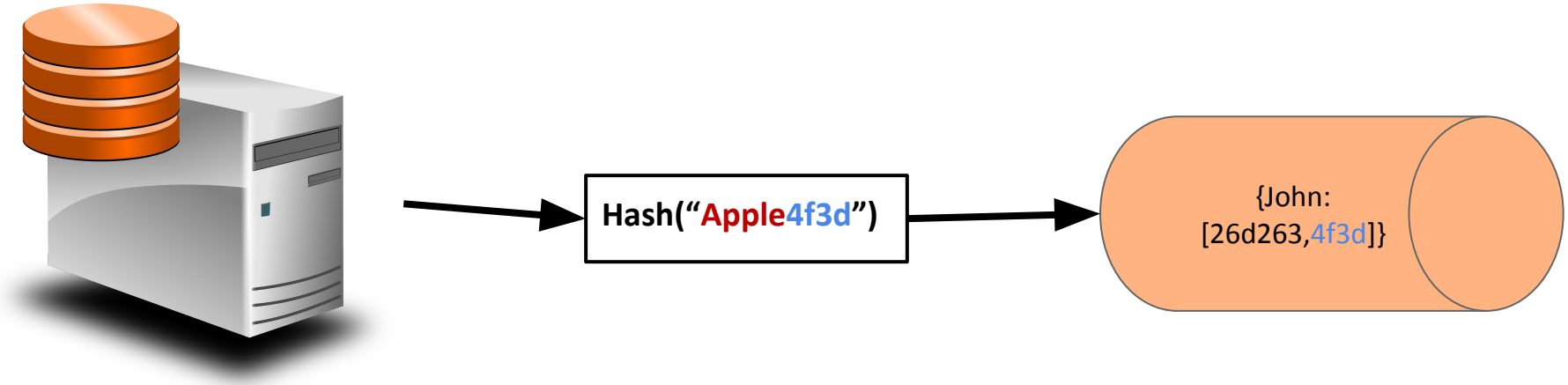
Hashing Passwords



Hashing Passwords

- We need a way to prevent commonly used passwords from hashing to the same value…
- Idea: Add a random value (a salt) to the end of every users password before hashing
  - Goal: Will practically make each hash random
  - Store the plaintext salt

Salting Passwords



Salting Passwords

Hash("Apple4f3d")

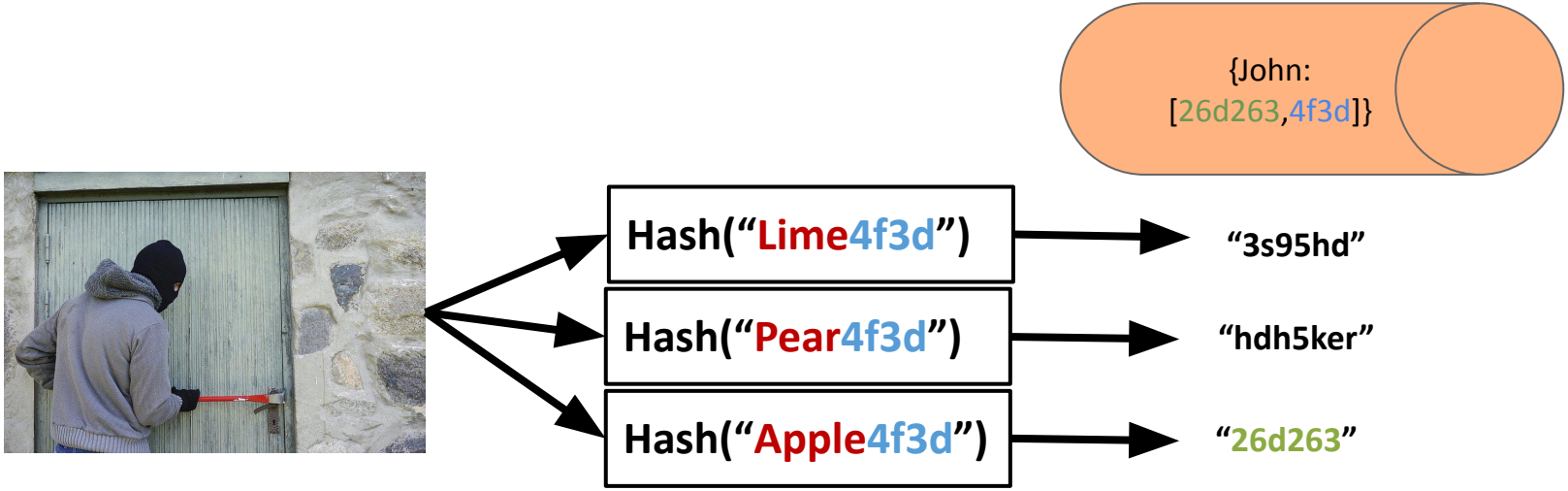
{John:
[26d263,4f3d]}

{.. 3dj4je..}

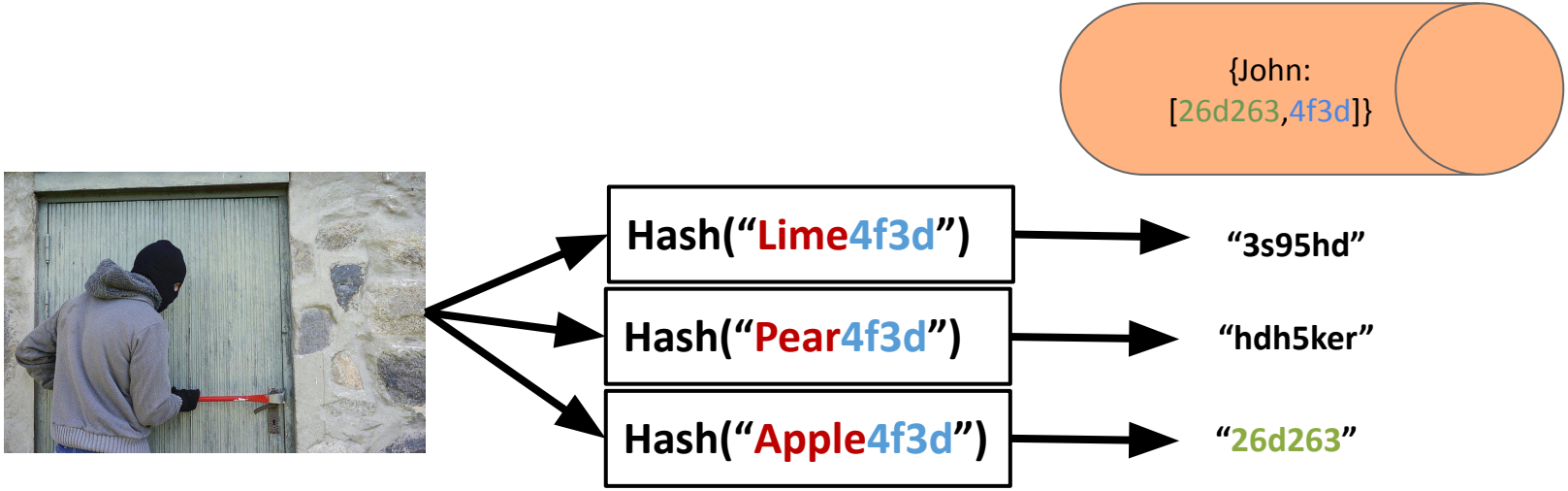
{.. d9sk3j..}

{.. 53kfa0..}

Salting Passwords



Salting Passwords



Attacker must now hash every guess with the stolen salt *for every user*