Diploma in Web Development – Part II

PHP Development – Week 4

PHP & Security

Presented by:
Julian Quirke
Web Development Educator

Week 3 Recap

Error Handling & Advanced Development

- Class Member Visibility
- Abstract Classes & Interfaces
- Error Handling in PHP

- Summary
- Q&A

Today's Lesson

PHP & Security

- Encrypted Data & HTTPS
- Data Validation with Hashing
- Storing Passwords Securely

- Summary
- Q&A

Let's Begin!

Encrypted Data & HTTPS

HyperText Transfer Protocol Secure

Encrypted Data & HTTPS

HyperText Transfer Protocol Secure

is the means of transferring hypertext over a computer network in an encrypted manner

Set Up a Secure Communication Channel

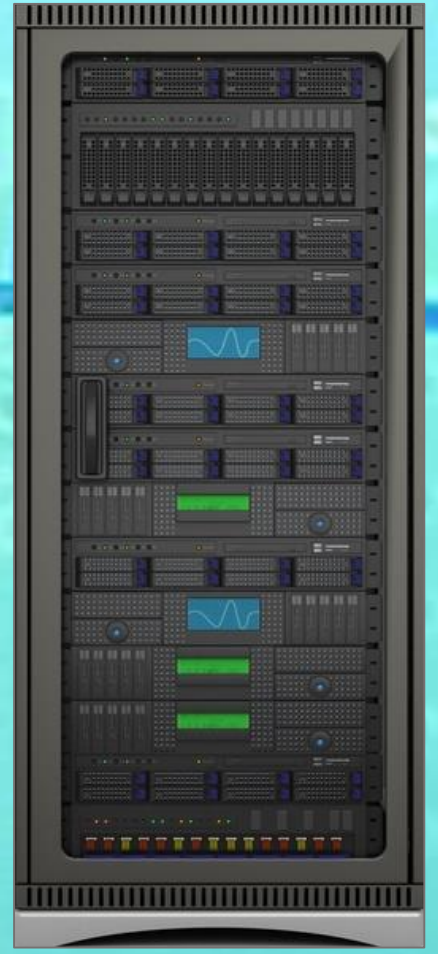
Connect to HTTP Site (Port 80)

Redirect to HTTPS Site

Connect to HTTPS Site (Port 443)

Provide Server Certificate

Secure Communication Begins



Encrypted Data & HTTPS

Why use HTTPS?

- Prevent “packet sniffing” of sensitive information
- Ensures information is actually sent from the expected user

Encryption

Encryption

is the process of encoding data in a manner that is only retrievable by authorised parties

Authorised parties are given a decryption key for this purpose

Encrypted Data & HTTPS

Encryption Terminology

SSL: Secure Sockets Layer

TLS: Transport Layer Security

HTTPS: Use of HTTP alongside TLS or SSL encryption

Encrypted Data & HTTPS

What Actually Happens in SSL/TLS

The end result of secure communication setup:

- User has a key
- Server has the opposite key

Information encrypted with one key can *only* be decrypted with the other key

- Only the server can decode messages from the user
- Only the user can decode messages from the server

Using a secure communication channel

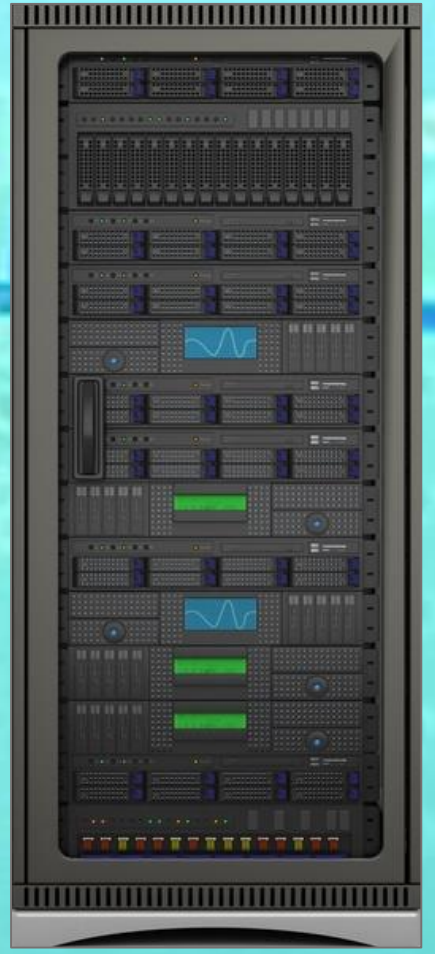
"I want pie"

"Here's some pie"

"more pie please"

"I have no idea what you just said"

Mallory
(aka the bad guy)



Using a secure communication channel

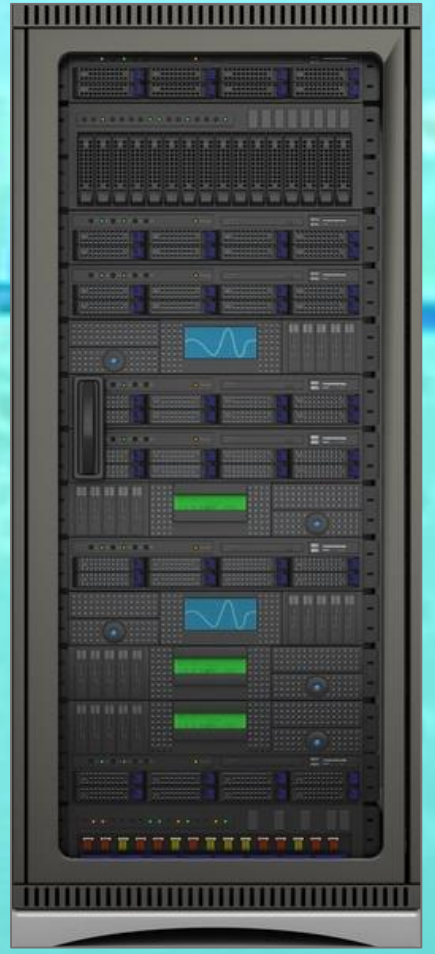
"I want pie"

"Here's some pie"

"Send me money."

"I don't understand, you're clearly an imposter."

Mallory
(aka the bad guy)



Using a secure communication channel

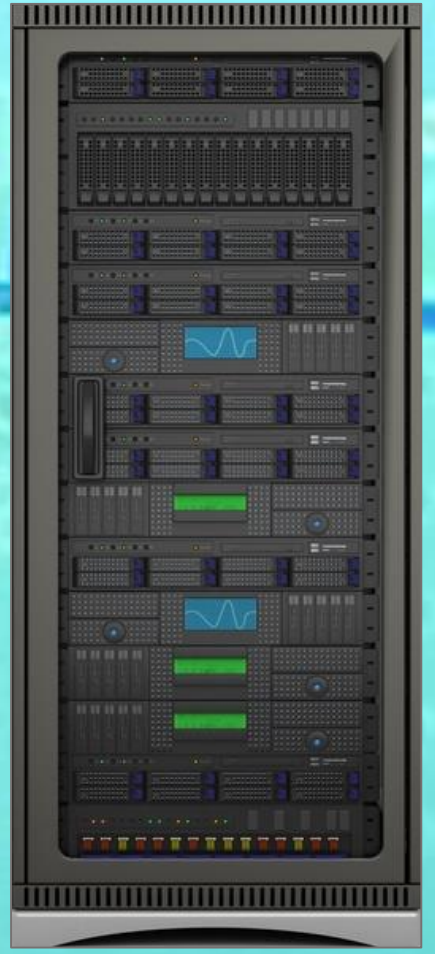
"I want pie"

"Here's some pie"

"I want to buy expensive things from you"

Mallory
(aka the bad guy)

"I don't understand: imposter!"



Using a secure communication channel

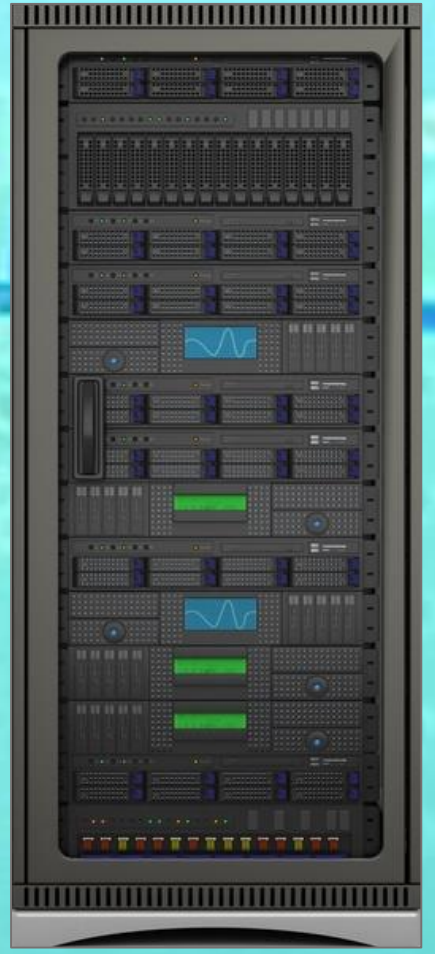
"I want pie"

"Here's some pie"

Mallory
(aka the bad guy)

"This is your email address"

"What did you say now?"



Data Validation

Data Validation

is the process of ensuring that data is clean, correct and useful.

Data Validation with Hashing

Data Validation in Programming

- Correct data type
- In the expected format
  - correct email format
  - correct array structure
  - etc
- Has the data been interfered with? ...

Data Validation with Hashing

Data Validation in Programming

Simple link to external Bootstrap stylesheet:

```
<link rel='stylesheet' href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css'
integrity='sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u' crossorigin='anonymous'>
```


Data Validation with Hashing

shawacademy

Data Validation in Programming

Hash Algorithm used

Hash of original file

Ensures that file has not been changed from original

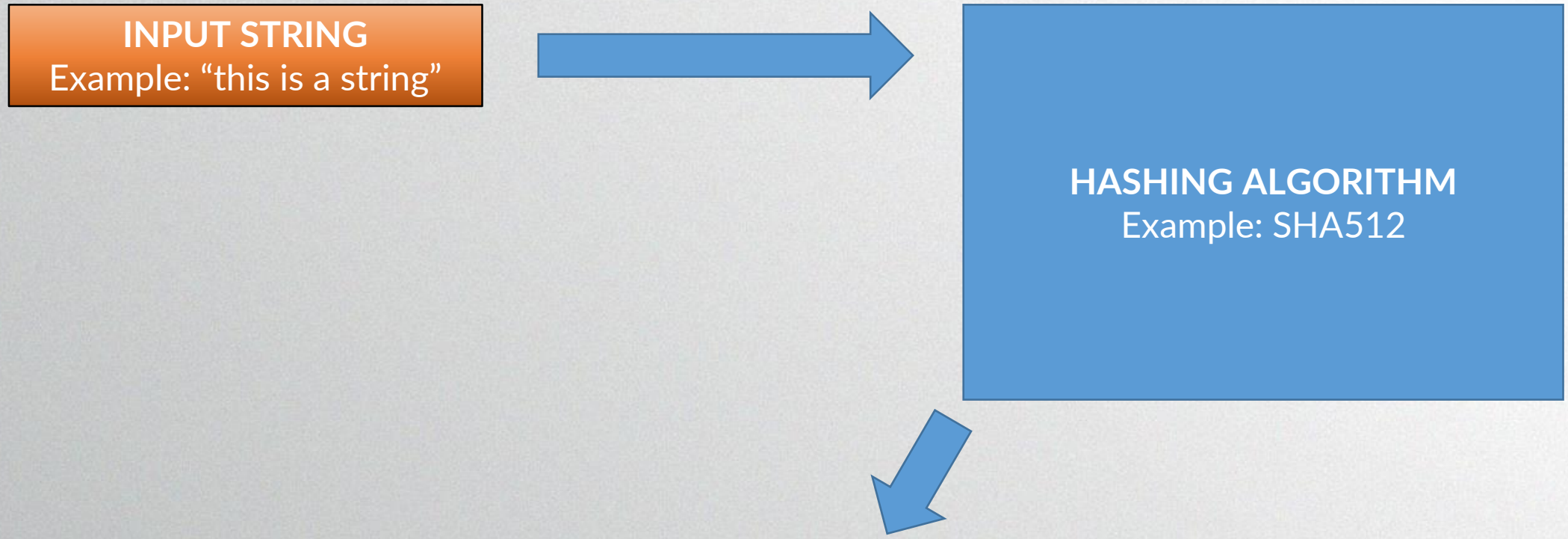
Hashing

Hashing

refers to the use of a cryptographic algorithm to convert data into a fixed length string of characters

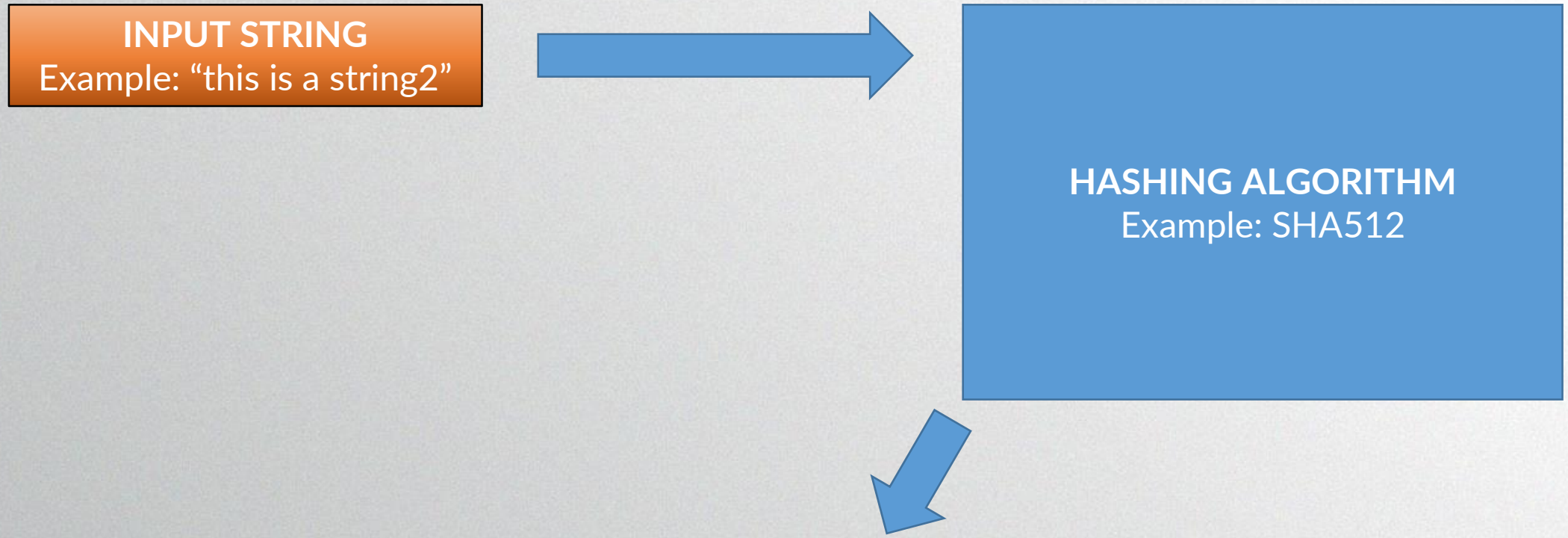
Data Validation with Hashing

Hashing Process



Data Validation with Hashing

Hashing Process



Data Validation with Hashing

Uses for Hashing

- Ensure information has not been changed
  - message authentication
  - the integrity of hosted content
- Store passwords in an unreadable format

Storing Passwords Securely

Storing Passwords Securely

Why use Hashing for Passwords?

- User privacy
- What happens if your database is compromised?
- Hacker cannot use these details to login to other applications

Storing Passwords Securely

Hashing Algorithms for Passwords

MD5 – now regarded as unsafe

SHA-1 – now regarded as unsafe against "well-funded opponents"

SHA-2(56)

SHA-5(12)

BCRYPT – adaptive password hashing

Storing Passwords Securely

BCRYPT

Maximum input password is 52 chars

Highly regarded as a safe, future proof

cryptographic algorithm

Brute-force resistance is very high

(Based on Blowfish)

Storing Passwords Securely

Salting

Storing Passwords Securely

Salting

is random data that is used as an additional input to a one way function that hashes a password

Storing Passwords Securely

Why Add a Salt?

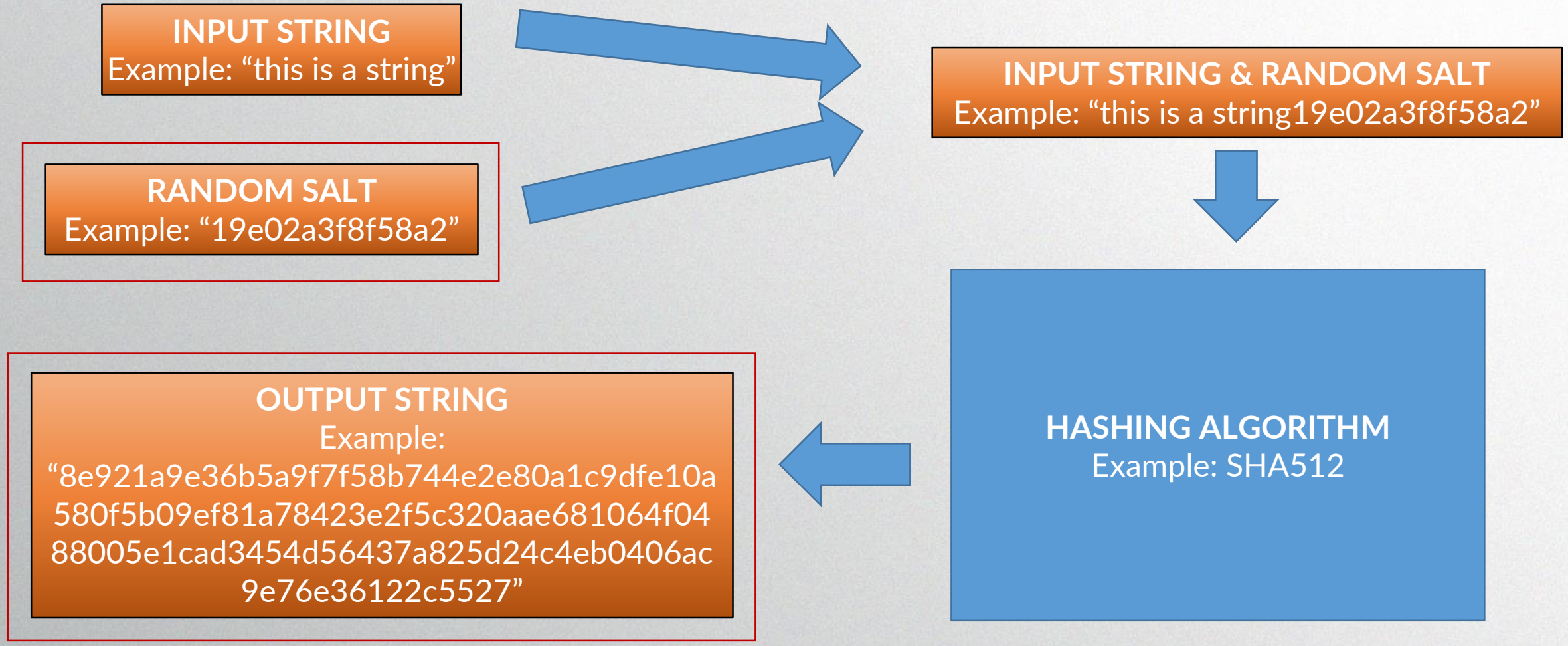
Helps defend against:

- Dictionary attacks
- Hash table attacks
- Rainbow table attacks

- Makes it *very very* unlikely that any two users will have the same hash stored as their password

Storing Passwords Securely

By their Powers Combined



Storing Passwords Securely

Review

1. Password entered as plain text
2. Random Salt added to password
3. Salted password is passed through hashing algorithm
4. Final output is a fixed length sequence of characters

Storing Passwords Securely

Final Note

- All password hashes are technically susceptible to brute force attacks
- BCRYPT has configurable difficulty

Next Week

Servers & Databases Semester

- The next session is "Relational Database Management Systems"
  - How Relational Databases Work
  - Field & Multi-Field Keys
  - Creating a Relational Database (with phpMyAdmin)

- Recordings are available within 24 hours after the live webinar
  - Go to www.shawacademy.com and then the Top Right Corner – Members Area

Q&A

Next Lesson is

Relational Database Management Systems

- Learn how data models are built using relational database management systems
- You will understand the relational model and the use of keys in development


www.shawacademy.com


www.facebook.com/shawacademy


www.twitter.com/shawacademy


support@shawacademy.com