

SAFE COMMUNICATION IN UNSAFE CONNECTIONS

In address bar of a browser, have you noticed either *http://* or *https://* at the time of browsing a website? If neither of these are present then most likely, it's *http://* Let's find out the difference...

In short, both of these are protocols using which the information of a particular website is exchanged between Web Server and Web Browser. But what's the difference between these two? Well, extra **s** is present in *https* and that makes it secure! What a difference A very short and concise difference between *http* and *https* is that *https* is much more secure compared to *http*.

Let us dig a little more.

Hyper**T**ext **T**ransfer **P**rotocol (HTTP is a protocol using which hypertext is transferred over the Web. Due to its simplicity, *http* has been the most widely used protocol for data transfer over the Web but the data (i.e. hypertext) exchanged using *http* isn't as secure as we would like it to be. In fact, hyper-text exchanged using *http* goes as plain text i.e. anyone between the browser and server can read it relatively easy if one intercepts this exchange of data. But why do we need this security over the Web? It's because you would not want some third party to intervene into our personal life and steal the data packets while your message was on the way to the receiver. And that's why *https* was introduced so that a secure session is setup first between Server and Browser. In fact, cryptographic protocols such as SSL turn *http* into *https* i.e. **https = http + cryptographic protocols**. To overcome this problem i thought of making something through which one can have the safest communication over internet through http only.



Solution proposed

Contents

1. Problem Inference
2. Solution
 - 2.1. Overview
 - 2.2. Parts of solution
 - 2.2.1. Encryption and Storing of Data
 - 2.2.1.1. Sign Up and Login Page
 - 2.2.1.2. Compose and Send messages
 - 2.2.1.3. Use of Javascript and AJAX
 - 2.2.1.4. Data Flow Diagram
 - 2.2.2. Working with Python and PHP
 - 2.2.2.1. Retrieving encrypted messages from Database in PHP
 - 2.2.2.2. Passing messages from PHP to Python
 - 2.2.2.3. Use of pythonanywhere.com
 - 2.2.2.4. Use of NLP in Decryption
 - 2.2.2.5. Returning original message to PHP
3. Technology Stack
 - 3.1. Source
 - 3.1.1. Any browser
 - 3.2. Frameworks
 - 3.2.1. Apache
 - 3.2.2. Flask
 - 3.2.3. W3.css
 - 3.3. Languages
 - 3.3.1. PHP
 - 3.3.2. SQL
 - 3.3.3. HTML
 - 3.3.4. JavaScript
 - 3.3.5. Python
4. Solution dependencies
 - 4.1. Advantages of solution
 - 4.2. Disadvantages of solution

1. Problem Inference

- This system will keep your messages safe in http enabled sites also.

From years, https extension is securing our websites from hackers, but when it comes to http, hackers attack here to steal your data. And data can be anything it may be your messages, or your bank account pin.

2. Solution

2.1 Overview

The goal of this system is to secure your data when passing in Transmission Control Protocol. And keep the data of the user safe while using any http server.

2.2 Parts of solution


To work on this problem, I came here with an interesting solution to encrypt the message in the client side before sending it to the server and decrypt it using Python NLP.

2.2.1. Encryption and Storing of Data:

In this part of Solution, we are going to do about half of our project which starts from Creating a account of a user to Storing the message into Database

2.2.1.1 Sign Up and Login Page

The sign up page is used as to register a new user in the Software and for the existing user they can use the Login Page. These page uses PHP as back-end programming. Sign up



form is at index.php and reg.do.php contains the registration of user back-end. And same like login.php contains the login form and back-end part is at login.do.php. After the successful login/signup user can reach the next page.

2.2.1.2 Use of JavaScript and AJAX

We are using AJAX to search users, for sending messages in search.php file. And after choosing user and entering the message user can submit the message and when user hit the submit button, a JavaScript function works which applies Caesar Cypher method with a randomly generated key value to encrypt the message.

2.2.1.3 Compose and Send messages

As user chose the receiver mail/name, and entered the message and after the encryption of message, encrypted message is transferred to work.php, which search about the recipient and id sends the encrypted message to the database.

2.2.1. Working with Python and PHP:

In the second part of Solution, we are going to do retrieve the encrypted message from Database and decrypt it using NLP

2.2.1.1 Retrieving encrypted message from database in PHP

We are retrieving the encrypted message from database which is checked for the user means the recipient is the current user.

2.2.1.2 Use of Pythonanywhere.com

PythonAnywhere is an online integrated development environment (IDE) and web hosting service (Platform as a service) based on the Python programming language. We are using Flask to run Python in web.

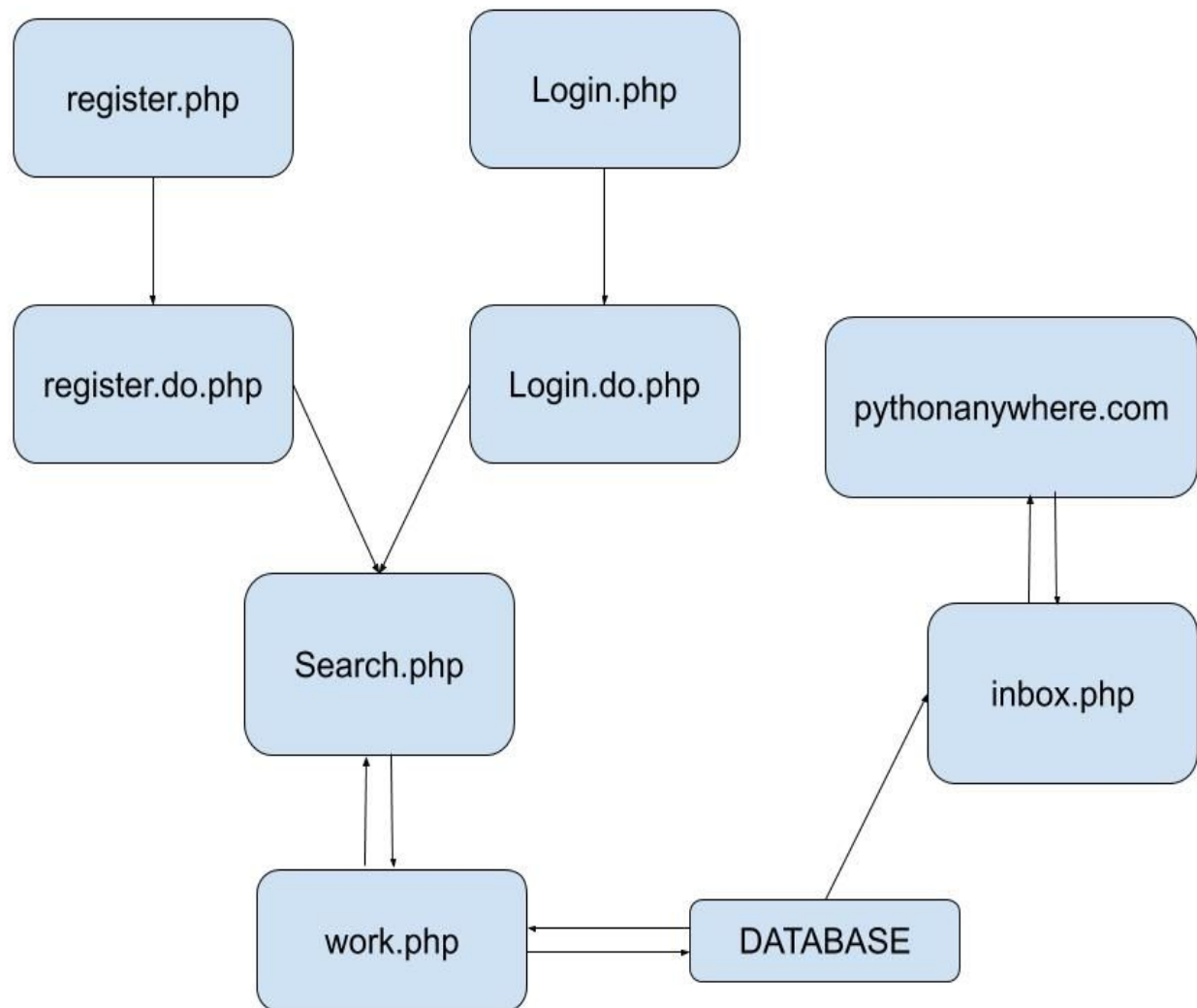
2.2.1.3 Passing message from PHP to Python

After retrieving the encrypted message from database, we are passing the message though the link of pythonanywhere.com. In this website, message is decrypted using NLP and without using a key value.

2.2.1.3 Passing message from Python to PHP

After message into link of pythonanywhere.com, the output is decrypted message and we can fetch it through `file_get_content()` function in PHP. And output the data in the inbox.php.

2.2.3. FLOW CHAR



T

2.2.3. Back-end Programming

2.2.3.1 index.php

```
<?php
    session_start();
?>
<!DOCTYPE html>
<html>
    <head>
        <title>Chat WEb</title>
        <link rel="stylesheet" href="css/w3.css" />
    </head>
    <body class="w3-green">
        <div class="w3-green" style="margin-top: 0px;">
            <div class="w3-card-8 w3-teal" style="margin-left:35%;margin-top:5% ;width:30%;height: 100%;">
                <div class="w3-margin-left w3-margin-right w3-center">
                    <br><h2 style="text-shadow: 2px 2px 2px black">Chat Group</h2>
                    <h5 style="text-shadow: 2px 2px 2px black">- Chatting portal </h5>
                    <form method="POST" action="reg.do.php">
                        <input type="text" placeholder="Enter your name" name="name" style="width:100%;"
class="w3-round w3-input w3-border w3-light-grey" /><p>
                        <input type="text" placeholder="Enter your email" name="email" style="width:100%;"
class="w3-round w3-input w3-border w3-light-grey" /><p>
                        <input type="password" placeholder="Enter your password" name="password"
style="width:100%;" class="w3-round w3-input w3-border w3-light-grey" /><p>
                        <button type="submit" class="w3-btn" style="width:100%;">Sign Up</button>
                    </form>
                    <a href="login.php" ><button class="w3-btn w3-blue" style="box-shadow: 2px 2px 5px black">Log
In</button></a>
                </div>
            </div>
        </div>
```

```

        </div>
    </div>
</body>
</html>

```

2.2.3.2 reg.do.php

```

<?php

include 'db.php';
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$name = $_POST['name'];
$email = $_POST['email'];
$password = $_POST['password'];
if($err == 0){
$sql = "INSERT INTO users (name, email, password) VALUES ('$name', '$email',
'$password')";

if ($conn->query($sql) === TRUE) {
    $_SESSION['email'] = $email;
    $_SESSION['id'] = $row['id'];
    $_SESSION['name'] = $row['name'];
    setcookie($email, $password, time() + (86400 * 30), "/");
    header("location: search.php");

} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
}
else{

```



```

        header ('location: reg.php');
    }
?>

```

2.2.3.3 login.php

```

<?php
    session_start();
?>
<?php
    if(isset($_SESSION['id'])){
        header("location: search.php");
    }
?>
<html>
    <head>
        <title> Chat </title>
        <link rel="stylesheet" href="css/w3.css" />
    </head>
    <body class="w3-teal">
        <div class="" style="margin-top: 0px;">
            <div class="w3-card-8 w3-green" style="margin-left:35% ;margin-top:10%;
width:30%;">
                <div class="w3-margin-left w3-margin-right w3-center">
                    <br><h2 style="text-shadow: 2px 2px 2px black">Chat Web</h2>
                    <h5 style="text-shadow: 2px 2px 2px black">- Chat h5</h5>
                    <form method="post" action="login.do.php">
                        <input type="text" placeholder="Enter your email" name="email"
style="width:100%;" class="w3-round w3-input w3-border w3-light-grey" /><p>
                        <input type="password" placeholder="Enter your password" name="password"
style="width:100%;" class="w3-round w3-input w3-border w3-light-grey" /><p>
                        <button type="submit" class="w3-btn" style="width:100%;">Log In</button>
                    </form>
                    <br>
                    <a href="index.php" ><button class="w3-btn w3-blue" style="box-shadow: 2px
2px 5px black;text-shadow: 2px 2px 2px black;">Registration</button></a>
                    <br><br>
                </div>
            </div>
        </div>
    </body>
</html>

```

```

        </div>
    </body>
</html>

```

2.2.3.4 login.do.php

```

<?php
    session_start();
    include("db.php");
    if($_SERVER["REQUEST_METHOD"] == "POST") {
        $email = $_POST['email'];
        $password = $_POST['password'];
        $sql = "SELECT id, name FROM users WHERE email = '$email' and password = '$password'";
        if($result = mysqli_query($conn,$sql)){
            $row = mysqli_fetch_assoc($result);
            $count = mysqli_num_rows($result);
            if($count == 1) {
                $_SESSION['email'] = $email;
                $_SESSION['id'] = $row['id'];
                $_SESSION['name'] = $row['name'];
                setcookie($email, $password, time() + (86400 * 30), "/");
                header("location: search.php");
            }else {
                header("location: login.php");
            }
        }
        else{
            echo mysqli_error($conn);
        }
    }
?>

```

2.2.3.5 aja.php

```

<?php
include 'db.php';
if(isset($_REQUEST["term"])){
    $sql = "SELECT * FROM users WHERE name LIKE ? OR email LIKE ? LIMIT 5";
    if($stmt = mysqli_prepare($conn, $sql)){
        mysqli_stmt_bind_param($stmt, "ss", $param_term, $param_term);

        $param_term = '%' . $_REQUEST["term"] . '%';
        if(mysqli_stmt_execute($stmt)){
            $result = mysqli_stmt_get_result($stmt);
            if(mysqli_num_rows($result) > 0){
                while($row = mysqli_fetch_array($result, MYSQLI_ASSOC)){
                    echo "<p><b>" . $row["name"] . "</b> <span style='color:grey'>("
.$row['email'].")</span></p>";
                }
            } else{
                echo "<p>No matches found</p>";
            }
        } else{
            echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
        }
    }
    mysqli_stmt_close($stmt);
}
mysqli_close($link);
?>

```

2.2.3.6 search.php

```
<?php
    session_start();
?>
<html>
<head>
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $('.search-box #per').on("keyup input", function(){
        /* Get input value on change */
        var inputVal = $(this).val();
        var resultDropdown = $(this).siblings(".result");
        if(inputVal.length){
            $.get("aja.php", {term: inputVal}).done(function(data){
                // Display the returned data in browser
                resultDropdown.html(data);

            });
        } else{
            resultDropdown.empty();
        }
    });

    // Set search input value on click of result item
    $(document).on("click", ".result p", function(){
        $(this).parents(".search-box").find('input[type="text"]').val($(this).text());
        $(this).parent(".result").empty();
    });
});
</script>
<script>
function myFunction() {
```

```
var letter = document.getElementById("work").value;
var min=1;
var max=62;
var alphabet =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
var key=Math.floor(Math.random() * (+max - +min)) + +min;
var text = "";
var data = "";
var i;
for (i = 0; i < letter.length; i++) {
    var index = alphabet.indexOf(letter[i]);
    if (index >=0 ){
        var pos = key + index;
        text += alphabet[pos%62];
    }
    else{
        text += letter[i];
    }
}
var text = text.replace(/ /g,"^");
document.getElementById("work").value = text;
}
</script>
<title>PHP Live MySQL Database Search</title>
<style type="text/css">
    body{
        font-family: Arail, sans-serif;
    }
    .search-box{
        width: 300px;
        position: relative;
        display: inline-block;
        font-size: 14px;
    }
    .search-box input[type="text"]{
        height: 32px;
        padding: 5px 10px;
        border: 1px solid #CCCCCC;
        font-size: 14px;
```

```

}
.result{
    position: absolute;
    z-index: 999;
    top: 100%;
    left: 0;
}
.search-box input[type="text"], .result{
    width: 100%;
    box-sizing: border-box;
}
/* Formatting result items */
.result p{
    margin: 0;
    padding: 7px 10px;
    border: 1px solid #CCCCCC;
    border-top: none;
    cursor: pointer;
}
.result p:hover{
    background: #f2f2f2;
}
</style>
</head>
<body>
WELCOME <?php echo $_SESSION['name']; ?>
<form method="get" action="work.php">
    <div class="search-box">
        <input id="per" type="text" autocomplete="off" placeholder="Search user....."
name="person" />
    <div class="result">
    </div>
<br><br><br><br><br>
    <textarea id="work" rows="4" cols="50" placeholder="Enter message"
name="mess"> </textarea>
    <button onclick="myFunction()" type="submit"> Send </button>

</form>
</body>

```



 </html>

2.2.3.7 work.php

```
<?php
    session_start();
include 'db.php';
$id = $_SESSION['id'];
$email = $_GET['person'];
$mess = $_GET['mess'];
$fir = strpos($email,"( ");
$las = strpos($email," )");
$mail = "";
for ($i=$fir+2; $i < $las; $i++) {
    $mail = $mail.$email[$i];
}
$sql = "SELECT id FROM users WHERE email = '$mail'";
$result = mysqli_query($conn,$sql);
$row = mysqli_fetch_assoc($result);
$count = mysqli_num_rows($result);
$rid = 0;
if($count == 1) {
    $rid = $row['id'];
}
else {
    echo "error1";
}
echo "Reci - ".$rid;
if ($rid !=0){
$id = $_SESSION['id'];
$sql = "INSERT INTO message (senderid, receiverid, mess) VALUES ('$id', '$rid', '$mess')";
    if ($conn->query($sql) === TRUE) {
        echo "Done";
    } else{
        echo "Error";
    }
}
```

```

    }
}
else{
    echo "No address found";
} ?>

```


2.2.3.8 inbox.php

```

<?php
session_start();
include("db.php");
?>
<!DOCTYPE html>
<html>
<head>
    <title> Chat Detector </title>
    <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
</head>
<body>
<table class="w3-table-all w3-large">
    <tr class="w3-green">
        <th>Name</th>
        <th>Message</th>
        <th>Time</th>
    </tr>
<?php
$myid = $_SESSION['id'];
$sql = "SELECT * FROM message RIGHT JOIN users ON message.senderid=users.id
WHERE receiverid='$myid'";
$result = mysqli_query($conn,$sql);
while ($row = mysqli_fetch_assoc($result) ){
    $link = "http://mayankgbrc.pythonanywhere.com/?data=".$row['mess'];
    $data=file_get_contents($link);
    echo
    "<tr><td>".$row['name']. "</td><td>".$data."</td><td>".$row['time']. "</td></tr>";
}

?>
</table>

```

```
</body>
</html>
```

2.2.3.9 db.php

```
<?php
$host = "sql107.unaux.com";
$user = "unaux_22180717";
$pass = "x16eva1dyho";
$db_name = "unaux_22180717_proj";
$conn = new mysqli($host, $user, $pass, $db_name);
?>
```

2.2.3.9 flask_app.py

```

from flask import Flask
from flask import request
import time
import nltk
from nltk.corpus import words
nltk.download('words')
import re
app = Flask(__name__)
@app.route('/')
def hello_world():
    wording = set(words.words())
    alphabet =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
    letter = request.args.get('data')
    org_data = ""
    maxi = 0
    for key in range(0,62):
        enc_data = ""
        word_token = []
        counter = 0
        for i in letter:
            if i in alphabet :
                enc_data = enc_data+alphabet[(key+alphabet.find(i))%62]
            else:
                enc_data = enc_data+i
        word_token = [i.lower() for i in enc_data.split("^")]
        word_token = [re.sub('[!@#$.%,;()""?{}]<>/:+=*&%`~1234567890', "", i) for i in
word_token]
        counter = sum([1 for i in word_token if i in wording])
        if(counter>maxi):
            maxi = counter
            org_data = enc_data
    new_data = ""
    for i in org_data:
        if i == '^':
            new_data = new_data + " "
        else:
            new_data = new_data + i
    return str(new_data)

```



Technology Stack

3. Technology Stack

A collection of various platforms, tools, frameworks and libraries comes into play while developing the solution for the problem at hand.

3.1 Source

3.1.1 Any Browser

It is expected that a major portion of the clients using this kind of model will be using any browser for their browsing purpose. Thus, we developed this chat detection website which can easily be browsed over different browsers in laptops , mobiles etc.

3.2 Frameworks

3.2.1 Apache

Firstly we used PHPmyadmin as our apache for testing purpose of the website then we hosted in at <https://zafire.in/>

3.2.2 W3CSS

For designing our websites we have used the w3css element of cascading style sheet from its predefined library called w3css

3.2.3 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.

3.3 Languages

3.3.1 PHP

Server scripting language widely used in Web Development.

3.3.2 SQL

Used for the purpose of making database of the chat records so that we can have some evidence against false chats or suspicious chats summary.

3.3.3 HTML

Used to make HTML web pages for the website and also for making forms and other required things we generally need while developing a websites.

3.3.4 CSS

Used for the designing purpose of the website and also for making website to look colorful.

3.3.5 JavaScript

Client side validation language.

3.3.6 Python

Python is a very powerful popular programming language.

Dependencies/Show Stopper

4. Solution Dependencies

4.1 Advantages of solution

- This application can be a great help to all the non-ssl hosted websites as ssl needs to be bought.
- Users can rely on this application and can use it to share their confidentials because the encryption key is set to random.

5. Disadvantages of solution

- If internet connection fails, this system won't work.
- For words that is not included in nltk.corpus we might face difficulty in scoring the accuracy of the algorithm which in turn can lead to disastrous decryption.

CONCLUSION:

This application is an encryption /decryption model where you can make secure communications over an unsafe connections such as http.. **https = http + cryptographic protocols**. Hence what this application does is, makes some cryptographic protocols

And secures our communication via internet for privacy concerns and reliable communications.



REFERENCES:

- <https://www.pythonanywhere.com/user/mayankgbrc/>(using flask through this platform)
- <https://www.fullstackpython.com/flask.html>(flask)
- <https://www.tutorialrepublic.com/php-tutorial/php-mysql-ajax-live-search.php>(use of ajax)
- <https://www.w3schools.com/html/default.asp>
- <https://www.w3schools.com/php/default.asp>