International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 1 ǁ 2021 ǁ PP. 16-22

Responsive Design in Web Development with Security Using Optimization Algorithms

P.Ajitha

Abstract:
Responsive design allows software developers to build a Web page that can dynamically adapt to the size of the devices. This development philosophy enables the rendering of Web pages in a fast and optimized way, ensuring a good user experience on mobile devices, tablet and desktop.The world is exceedingly reliant on the Internet. Nowadays, web security is biggest challenge in the corporate world. It is considered as the principle framework for the worldwide data society. Web applications are prone to security attacks. Web security is securing a web application layer from attacks by unauthorized users. A lot of the issues that occur over a web application is mainly due to the improper input provided by the client.
Hashing algorithms are commonly used to convert passwords into hashes which theoretically cannot be deciphered. This paper analyses the security risks of the hashing algorithm MD5 in password storage and discusses different solutions, such as salts and iterative hashing. We propose a new approach to using MD5 in password storage by using external information, a calculated salt and a random key to encrypt the password before the MD5 calculation. We suggest using key stretching to make the hash calculation slower and using XOR cipher to make the final hash value impossible to find in any standard rainbow table.
Keywords- *MD5, Web development; Responsive design; Web designing; User experience ,Hash Function, Security Standards.*

Date of Submission: 29-12-2020 Date of acceptance: 10-01-2021

I. INTRODUCTION

A modern website is a tool for any company to increase its visibility towards potential customers. It is common for companies, institutions, organizations and individuals to have websites to reach audience or customers. However, it is not enough just to be present on web and available through web search engines anymore. People are spending most of their time online and most of them are using handheld devices to access the Internet, so now websites need to be optimized for all these devices in order to provide the best user experience.

While some web technologies are outdated (e.g. Flash) due to their incompatibility with different versions of the browser, new technologies, and www standards are in, the need of HTML5 and CSS3 for responsive designing of website is increasing rapidly.

Web security is an important aspect for web applications. Today web security is a real concern related to the Internet. It is considered as the principle framework for the worldwide data society. Web applications provide a better interface for a client through a web page .The web page script gets executed on client web browser. Web applications, needing user authentication, typically validate the input passwords by comparing them to the real passwords, which are commonly stored in the company's private databases. If the database and hence these passwords were to become compromised, the attackers would have unlimited access to these users' personal. Nowadays, databases use a hash algorithm to secure the stored passwords but there are still security breaches.

II. NEED FOR RESPONSIVE WEB DESIGN

In the past few years, the use of smartphones has grown rapidly. In 2014, only 22% turned to their phone first to browse the internet. Only 2/3rds of people even owned a smartphone. Since then, the roll-out of 4G internet and other innovations impacted the way people used mobile devices; throughout 2014, 4G subscriptions skyrocketed from 2.7 million to 23.6 million.

By the end of 2016, mobile web usage had overtaken desktop for the first time, which just goes to show how many people browse on smartphones and tablets instead of a laptop or desktop computer. This highlighted a dramatic shift in behaviour; with the UK using their mobile devices to shop online, carry out online banking and scroll through social media. Ofcom reported at the time, that 7 in 10 adults use a smartphone, with over-65s now more likely to use one too, as the public becomes even more tech-savvy.

For these reasons alone, it became hugely important to invest in responsive web design ; your business needs a website that works well on a smartphone or other mobile device screens, due to the sheer amount of people who could be using your website through a smartphone or tablet.

How to Make a Responsive Design

There are a number of things to think about when creating a responsive layout. It is a process that requires a design system and hierarchy of content across devices.

The three main components of a responsive web design include:

- A fluid grid
- Flexible text & images
- Media queries

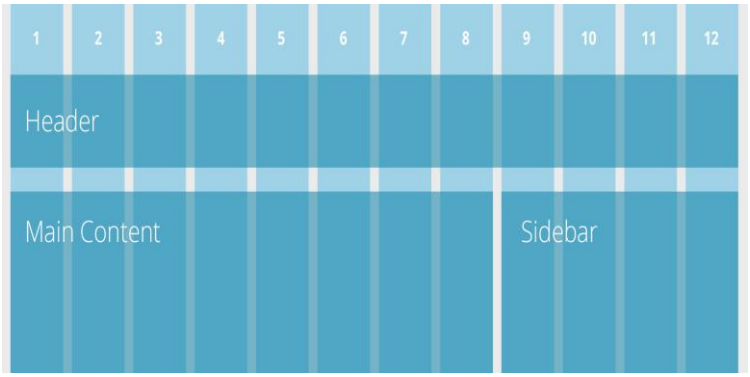
A Fluid Grid

The grid is a crucial element for creating a responsive layout.

Now, grids aren't anything new.

Web designers have used grids for building websites since the beginning. However, in the past, these grids were fixed width and didn't lend themselves to support a fluid website layout.

A fluid grid used for responsive sites will ensure that the design is flexible and scalable. Elements will have consistent spacing, proportion, and can adjust to a specific screen-width based on percentages.



The most common screen sizes for responsive design are:

Large **desktop**
1220 px and more
Desktop
960 – 1219 px
Tablet **(Portrait)**
768 – 959 px
Mobile **(Wide)**
480 – 767 px
Mobile
479 px and less

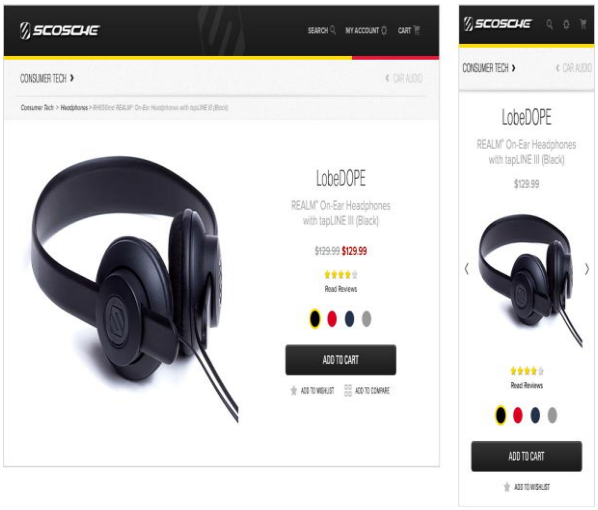
With a fluid grid, users will have the best experience on whatever screen they view your responsive website on.

Flexible Text & Images

The way to display text varies depending on what device a user is viewing your site on, but it should be readable *no matter what*. On mobile responsive websites, there is an opportunity to increase font size and line height (the spacing between each line of text) for legibility.

Flexible text and images adjust within a website layout width, according to the content hierarchy set with the CSS (stylesheet). Text can now be legible regardless of the end user's device. With a flexible container (within the grid), text can wrap with an increase in font size on smaller devices.

Flexible images can prove to be more challenging because of load times on smaller device browsers. But these images can scale, crop, or disappear depending on what content is essential to the mobile experience.



Media Queries

This is code that powers the flexibility of a layout on responsive websites. Media queries specify the CSS to be applied accordingly, depending on a device's breakpoint (for example, iPhone portrait orientation or iPad landscape orientation, etc.).

Media queries allow for multiple layouts of a design, which use the same HTML-coded web page.

```
/* Portrait and Landscape */
@media only screen
  and (min-device-width: 320px)
  and (max-device-width: 480px)
  and (-webkit-min-device-pixel-ratio: 2) {

}
```

BASE 64

Base64 is a part of a group of binary-to-text encoding schemes which represent binary data in an ASCII string format by translating it into a radix-64 format.

Why do we need to use Base64?

Computer understand and communicate in 0s and 1s - Binary. However there is a need to send rich data likes images, audios, videos. So as to transfer this kind of rich data between the computers it needs to encoded to 0s and 1s first and need to send across. Destination computer needs to decoded the data. There are many ways to encode and decode the data. It would be easy that all are agree for universal encoding. But that's not happening.

- There are lots on encoding created by many people and organizations. Later on ASCII became the universal standards wit 7 bits per character. But computer stores 8 bits of binary data per character. So ASCII is not suitable to transferring 8 bits of data.
- Base64 encoding was invented and introduced to solve this problem. Base64 encoding helps to encode bytes to bytes which is very safe to transfer without losing or corrupting.

Disadvantages Base64 :

- Though Base64 increases performance but be careful. Doing so will increase image size approximately 20-25%. than what it is actually in its binary form. So more data has to be transferred on internet. For mobile devices it is a bit drawback.
- Even if we apply gzip compression, doing so will only decrease css file size to around 10-12%.
- IE6 & IE7 not supports Data URI which means base64 images will not be loaded in ie6 & ie7 browsers.

- If apply base64 encoding to lots of medium sized images it will increase HTML content size or CSS content size. So browser has to do roundtrip to get full content.

There is lot of disadvantages in Base64 encoding and decoding.
MD5 algorithm is a best algorithm for encoding and decoding.

HASH FUNCTION

Hash functions were introduced in cryptography in the late seventies as a tool to protect the authenticity of information. A hash function is a one-way encryption function that takes a variable-size input plaintext m and generates a fixed size hash output. It is computationally hard to decipher the hash and any attempt to crack it is practically infeasible. A "secure" hash function should be able to resist pre-image attacks and collision attacks. Due to the pigeonhole principle and birthday paradox, there will be some inputs that will produce the same hash result. The output length is of fixed size 128 bits, making a total of 2128 possible output hash values.

A cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the "message", and the hash value is sometimes called the message digest or simply digests.

MD5 Algorithm

A message digest is a code which is created algorithmically from a file and represents that file uniquely. If the file changes, then the message digest changes. In addition to allowing us to determine if a file has changed, message digests can also help to identify duplicate files. The Message Digest is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA. Other applications of message digest include E-mail security, cyclic redundancy Checksum for files on a network etc.

Steps:

1.The message "M" is padded so that its length in bits is similar to 448 modulo 512, that is, the padded message is less than 64 bits of multiple of 512.
2.Firstly, the padding consists of a single 1 bit in the first column, followed by enough zeros to pad the message to the required length till the 512 bit. Padding is always used, even if the original length of M happens to equal 448 mod 512. As a result, there is at least one bit of padding, and at most 512 bits of padding. Then the length in bits of the message uses before padding is appended as a 64-bit block.
3.The padded message is a multiple of 512 bits and, it is also a multiple of 32 bits.
4.Let M be the required message and N is the number of 32-bit words used in the padded message.
5.Due to the actual padding, N is a multiple of 16 bit.

Problem Definition:

The message can be a string of any variable length containing alphabets (both lower case and upper case), digits and special symbols (containing line feeds, form feeds and escape characters). In other words we can write a general formula for the message as:

Message = ((a-z) + (A-Z) + (0 - 9) + (! - ;)) *

The message digest should be a string of fixed length containing all the alphabets and special symbols.
Message Digest= ((a-z) + (A-Z) + (0 - 9) + (! - ;)) n
, Where n= a fixed natural number

Salting

A salt is a secondary piece of information made of a string of characters which are appended to the plaintext and then hashed. Salting makes passwords more resistant to rainbow tables as the salted hashed password will have higher information entropy and hence much less likely to exist in pre-computed rainbow tables. Typically, a salt should be at least 48 bits. Salting can be implemented using the following ways:
1) Single salt for all passwords: Given that the salt is sufficiently long and complex, a standard rainbow table, cannot be used to decipher the salted hashes. However, two same passwords will still produce the same hash.
2) Different random salt for each password and storing the salt within the database itself: If we use different salts for each password, two same passwords will have different hashes. The attacker has to generate different rainbow tables for each individual user, making it computationally harder for an attacker to crack the hashes. These salts can be stored in plaintext in the database. The purpose of the salt is not to be secret, but to be random enough to defeat the use of rainbow tables.

3) Use an existing column value: An existing column value like username can be used as salt. This solution is similar to the second solution discussed above. It defeats the use of a standard rainbow table, but a hacker might generate a rainbow table for a specific username, for example, "root" or "admin".

4) Use a variably located calculated salt: The salt location can be prefix (salt appended in front of password), infix (salt appended within the password) or postfix (salt appended at the end of the password). If the salt's location is made random, then cracking the passwords is made harder. For example, we can set the salt location to be equal to the password's length modulo 3. The salt can be calculated by using a random character sequence (stored in the database) and using other characters (embedded within the code). For example, the salt can be made to be a combination of the first two letters of username, random sequence of characters and the last 2 letters of username.

5) Use a variably located calculated salt including information outside the database and the application code: The hacker now has to break into the database and the server containing the application code. He also needs to obtain the additional information needed to crack the password. C. Improvement on MD5 processing The following methods can be used to improve the MD5 processing: 1) Improved hash function: The hash computation function is altered, for example using one of the following functions as shown below.

hash = Hash (password + salt)

hash = Hash (Hash (password) + salt)

hash = Hash (password + salt + key)

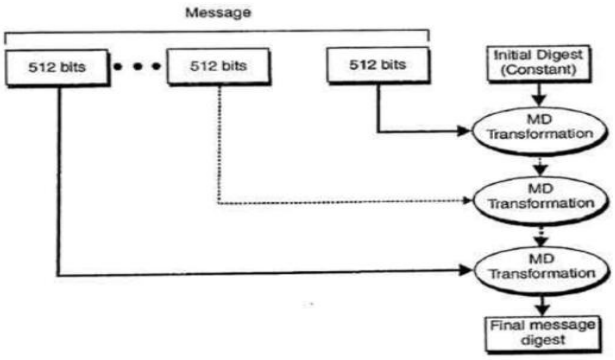
2) Iterative hashing: The password is hashed a number of times. MD5 is a fast hashing function, that is, it is computationally fast to calculate. Iterative hashing makes the calculation slower, hence computationally slower and more difficult to crack. The number of iterations can typically be made to be equal to 1000.

3) Key stretching: This makes a password more resistant to pre-computation attacks by making the attack workload bigger. Iterative hashing is used, where a weak key is fed into the hash algorithm and the output results in a stronger key.

Simple Key stretching: Only the key is hashed iteratively, as in

(4). No salt is involved.

key = Hash (key)



Architecture: Generation of Message Digest using MD5

How MD5 works

Preparing the input

The MD5 algorithm first divides the input in **blocks** of 512 bits each. 64 Bits are inserted at the end of the last block. These 64 bits are used to record the length of the original input. If the last block is less than 512 bits, some extra bits are 'padded' to the end.

Next, each **block** is divided into 16 **words** of 32 bits each. These are denoted as M_0 ... M_{15}.

MD5 Helper Functions

The Buffer

MD5 uses a buffer that is made up of four **words** that are each 32 bits long. These words are called A, B, C and D. They are initialized as

```
word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10
```

The Table

MD5 further uses a table K that has 64 elements. Element number i is indicated as K_i. The table is computed beforehand to speed up the computations. The elements are computed using the mathematical sin function:

$K_i = abs(sin(i + 1)) * 2^{32}$

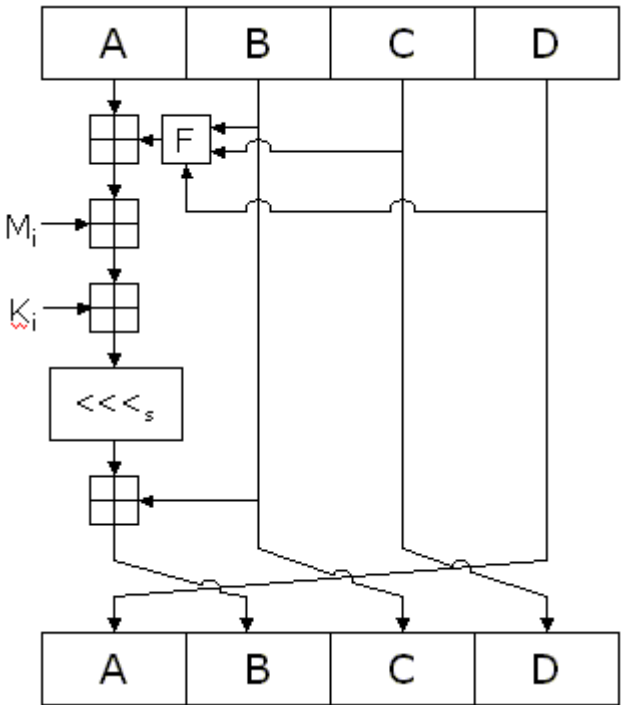
Four Auxiliary Functions

In addition MD5 uses four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word. They apply the logical operators and, or, not and xor to the input bits.

```
F(X,Y,Z) = (X and Y) or (not(X) and Z)

G(X,Y,Z) = (X and Z) or (Y and not(Z))

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X or not(Z))
```

Processing the blocks

The contents of the four buffers (A, B, C and D) are now mixed with the words of the input, using the four auxiliary functions (F, G, H and I). There are four *rounds*, each involves 16 basic *operations*. One operation is illustrated in the figure below.



The figure shows how the auxiliary function F is applied to the four buffers (A, B, C and D), using message word M_i and constant K_i. The item "<<<s" denotes a binary left shift by *s* bits.

Calculate MD5 Hash Value in PHP

```
<!DOCTYPE html>
<html>
<body>

<?php
$str = "Hello";
```

```
echo md5($str);
?>

</body>
</html>
```

Output

8b1a9953c4611296a827abf8c47804d7

Benefits of Responsive Design

1) Attract a wider audience
2) Easier to monitor analytics
3) Easier to maintain
4) Lower bounce rates
5) Consistency in design and brand

III. CONCLUSION

Responsive web design adapts the web page to different screen sizes and it is also prepared for the future-devices that haven't been released yet. Along with greater number of mobile devices, the importance of responsive web design is also increased. It is important for a business or a commerce website to be optimized for optimal user experience without major change in the flow, navigation and content. Implementation of responsive web design can result with greater number of visitors, increased sales and customer's satisfaction.

This research paper provides a complete survey of current research results under web application security. We have covered all properties of web application development, understood the important security functions and properties that secure web applications.

Password storage security is one important aspect of data security as most systems nowadays require an authentication method using passwords. Hashing algorithms such as MD5 are commonly used for encrypting plaintext passwords into strings that theoretically cannot be deciphered by hackers due to their one-way encryption feature

REFERENCES

[1]. Almeida, F., & Monteiro, J. (2017). Approaches and Principles for UX web experiences. International Journal of Information Technology and Web Engineering, 12(2), 49-64.
[2]. Alqahtani, A., & Goodwin, R. (2012). E-commerce smartphone application. International Journal of Advanced Computer Science and Applications, 3(8), 54-59
[3]. Berson, Thomas A. "Differential Cryptanalysis Mod 232 with Applications to MD5". EUROCRYPT. pp. 71–80. ISBN 3-540-564136. , 1992.
[4]. Xiaoyun Wang; Hongbo Yu. "How to Break MD5 and Other Hash Functions". EUROCRYPT. ISBN 3-540-25910-4, 2005.
[5]. Baker, S. (2014). Making it work for everyone: HTML5 and CSS level 3 for responsive, accessible design on your library's web site. Journal of Library & Information Services in Distance Learning, 8(3-4), 118-136.
[6]. Litayem, N., Dhupia, B., & Rubab, S. (2015). Review of cross-platforms for mobile learning application development. International Journal of Advanced Computer Science and Applications, 6(1), 31-39.
[7]. Rivest, R. The MD5 message-digest algorithm. RFC 1321, 37 (April 1992).
[8]. Zhang Shaolan, Xing Guobo, Yang Yixian, Improvement and Security Analysis on MD5 [J]. Computer Application, 2009, vol. 29(4):947-949.