



Sudan University of Science and Technology

Collage of Graduate Studies



Preserving Data Privacy in Cloud Computing

الحفاظ على خصوصية البيانات في الحوسبة السحابية

**A Thesis Submitted in Partial Fulfillment of the Requirements
of M.Sc. in Information Technology**

Prepared by: -

Yassir Eltayeb GadElseed Hassan

Supervised by: -

Dr. Faisal Mohammed Abdallah

November, 2017

الآية

قال تعالى:

(وَيَسْأَلُونَكَ عَنِ الرُّوحِ قُلِ الرُّوحُ مِنْ أَمْرِ رَبِّي وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا)

صدق الله العظيم

سورة الإسراء - الآية 85

الحمد لله

الحمد لله أقصى مبلغ الحمد.. والشكر لله من قبل
ومن بعد الحمد لله عن سمع وعن بصر.. الحمد لله عن
عقل وعن جسد الحمد لله عن ساق وعن قدم.. الحمد لله
عن كتفي وعن يدي الحمد لله عن قلبي وعن رئتني..
الحمد لله عن كلتي وعن كبدي الحمد لله عن أمي وعن
أبتي.. والحمد لله عن أخوات ذا العبد.

الحمد لله رب العالمين الذي جعل لكل شيء قدراً،
وجعل لكل قدر أجلاً، وجعل لكل أجل كتاباً.

الحمد لله ربّ العالمين، الذي سبّحت له الشمس
والنجوم الشهاب، وناجاه الشجر والوحش والدّواب،
والطّير في أوكارها كلُّ له أواب، فسبحانك يا من إليه
المرجع والمآب.

DEDICATION

Who taught me how to take the science of wealth ... To whom I try to
achieve a dream as long as he sees ...

To who gave me the fruits of his life ... To who taught me patience ...

My Father ...

To the big Heart ... To the rivers of tender tenderness ... And tried to grow
up... To who will the heaven rise under their feet ... To who was her prayer
the secret of our progress

My Mother ...

To those who were almost ready to be apostles ... To those who enlightened
our way with their knowledge ...

My distinguished teachers...

To those who joined us the path ... To whom we met with them without
time were the sweetest memories...

My Friends...

To those who dream together ... To those who share the sweetness of life
and happiness ... To whom we wish all beautiful ...

My Brothers...

ACKNOWLEDGEMENT

*First of all, all thanks belong to **ALLAH**, the almighty for giving me the will power to make this work; truly without his grace nothing is achievable.*

*I am extremely thankful to my respective guide **Dr. Faisal Mohammed Abdallah** for his valuable guidance, advice, motivation, encouragement, moral support, sincere effort.*

*My grateful sincere respect goes to my teacher **Dr. Hoida Ali Abdalgadir Ahmed**, who guides and advises me to make my first steps.*

*And my grateful to whom always encourage me from the beginning till the completion of the thesis my beloved **Mother and Father**.*

Finally, I would like to thank everyone who participates in success of this thesis.

ABSTRACT

Cloud computing is the result of the evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take advantage of all these technologies, without the need for deep knowledge or expertise with each one of them.

Security and privacy are among the top concerns standing in the way of wider adoption of cloud. In cloud computing the main concern is to provide the security to end user to protect files or data from unauthorized user. Security is the main intention of any technology through which unauthorized intruder can't access your file or data in cloud. In order to help to encrypt and decrypt the file at the user side that provides security to data at rest as well as while moving, an Algorithm has been designed. In this research, md5 Encryption Algorithm has been used to generate key. The algorithm depends on the key, and the key is generated from the user ID, each user has a different value than the other. This leads to a powerful algorithm, because each user has a different ID than the other.

المستخلص

الحوسبة السحابية (Cloud computing) هي نتيجة لتطور واعتماد التكنولوجيات والنماذج القائمة, والهدف من الحوسبة السحابية هو السماح للمستخدمين بالاستفادة من جميع هذه التقنيات, دون الحاجة إلى معرفة عميقة أو خبرة مع كل واحد منهم.

الأمن والخصوصية من بين أهم المخاوف التي تقف في طريقها على نطاق أوسع في إقرار الحوسبة السحابية , في الحوسبة السحابية الاهتمام الرئيسي هو توفير الأمن للمستخدم لحماية الملفات أو البيانات من المستخدمين غير المصرح لهم , والأمن هو الهدف الرئيسي لأي تقنية من خلالها لا يمكن للدخيل غير المصرح به الوصول إلى الملفات الخاص بك أو البيانات في الحوسبة السحابية. لذلك تم تصميم خوارزمية يمكن أن تساعد في تشفير وفك التشفير لملف المستخدم وتوفر الأمن للبيانات أثناء التحرك. في هذا البحث استخدمنا خوارزمية التشفير md5 لتوليد المفاتيح، الخوارزمية تعتمد على المفتاح (Key)، والمفتاح مولد من معرف المستخدم (User ID)، وكل معرف مستخدم له قيمة مختلفة عن الأخرى مما يؤدي إلى قوة الخوارزمية ، لأن كل مستخدم لديه معرف مختلف عن الآخر.

Contents

CHAPTER 1 INTRODUCTION

1.1	Introduction.....	1
1.2	Problem Statement:	2
1.3	Objectives:.....	2
1.4	Significance of the Study:.....	2
1.5	Methodology:.....	2
1.6	Hypothesis:.....	2
1.7	Scope:.....	3
1.8	Layout:.....	3

CHAPTER 2 BACKGROUND AND LITERATURE REVIEW

2.1	Introduction.....	4
2.2	Cloud computing types:.....	5
2.2.1	Public cloud:	5
2.2.2	Private cloud (INTERNAL OR EXTERNAL):.....	6
2.2.3	Community cloud:	7
2.2.4	Hybrid cloud:	7
2.3	Confidentiality and privacy:	7
2.4	Previous Studies:	8

CHAPTER 3 METHEDOLOGY, TOOLS AND TECHNIQUES

3.1	Algorithms:	13
3.2	Generate Key function:.....	13
3.3	Encrypt function:	13
3.4	Ensmall Function:	14
3.5	Decryption Function:	16
3.6	PHP:.....	16
3.7	Message Digest 5 (MD5):.....	17

CHAPTER 4 Implementation

4.1 Introduction..... 16
4.2 System Screens:..... 16
4.3 RESULTS: -..... 20

Chapter 5 Conclusion and Recommendations

5.1 Conclusion: 23
5.2 Recommendations:..... 23
5.3 Reference: 24
5.4 Appendices:..... 24

LIST OF FIGURES:

Figure Number	Description	Page No.
Figure 1	login form	18
Figure 2	Sign Up Form	19
Figure 3	attachment Form	20
Figure 4	Encrypt or decrypt Button.	21
Figure 5	upload files	22

LIST OF TABLES:

Table Number	Description	Page No.
Figure 5.1	Summary of related works	12

LIST OF TERMS:

Term	Description
IT	Information Technology
SOA	Service Oriented Architecture
User Id	User Identifier
md5	Message Digest 5
ASCII	American Standard Code for Information Interchange
CSP	Cloud Service Providers
IBM	International Business Machines
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
SQL	Structured Query Language
AWS	Amazon Web Services
S3	Simple Storage Service
SQS	Simple Queue Service
EC2	Elastic Compute Generation 2
EAP-CHAP	Extensible Authentication Protocol- Challenge Handshake Authentication Protocol
OTP	One Time Password
S-Box	substitution-box
DSA	Digital Signature Algorithm
AES	Advanced Encryption Standard
ASP	Active Server Pages
RSA	Rivest Shamir Adleman
3DES	Triple (Data Encryption Standard)
PHP	Personal Home Page

CHAPTER I
INTRODUCTION

1.1 Introduction:

Cloud computing is the result of the evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take advantage of all these technologies, without the need for deep knowledge or expertise with each one of them. The cloud aims to cut costs, and help the users focus on their core business instead of being impeded by IT obstacles.

The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing device into one or more virtual devices, each of which can be easily used and managed to perform computing tasks. With operating system–level virtualization essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provide resources on demand. By minimizing user involvement, automation speeds up the process, reduces labor costs and reduces the possibility of human errors. Users routinely face difficult business problems. Cloud computing adopts concepts from Service-oriented Architecture (SOA) that can help the user break these problems into services that can be integrated to provide a solution. Cloud computing provides all of its resources as services and makes use of the well-established standards and best practices gained in the domain of SOA to allow global and easy access to cloud services in a standardized way[1].

1.2 Problem Statement:

Security is an important issue for lots of our life aspects. In Cloud Computing it's particularly needed so as to maintain data, but a critical problem is that if the key is known, the data or the documents will be seen easily and attacked as well.

1.3 Objectives:

Designing an algorithm so as to protect the original text from the unauthorized users.

Generate unique keys for encrypt in the system.

1.4 Significance of the Study:

It appears in the aspect of which data will be more secure and private after applying the suggested algorithm.

1.5 Methodology:

Create a function to receipt the user ID and encrypt it using md5 algorithm, leads to generate a code consists of 32 bit numbers and letters and splits the code to take just the numbers and creates six keys by using ASCII (American Standard Code for Information Interchange) code technique.

Encrypt the text by ASCII code technique and insert the encrypted key to encrypt the text.

1.6 Hypothesis:

Using keys which generated from user ID using MD5 algorithm and ASCII code technique increases the strength of the algorithm.

using ASCII code technique plus an encrypted key leads to difficulty of finding the original text.

1.7 Scope:

The suggested algorithm could be applied in the drop-box application.

1.8 Thesis Layout:

Chapter one gives an introduction about the project, defining the problem, objectives, significance of the Study, methodology, hypothesis and scope. Chapter two contains two parts. Part one represents a general background about cloud computing, platforms, types, services and security problem. Part two is the related studies and techniques that used in cloud computing. Chapter three also contains two parts. The first part explains the tools and techniques used in this project. The second part is the UML design for the project functionality. Chapter four contains the project implementation. Chapter five is the results and recommendations.

CHAPTER II
BACKGROUND AND LITERATURE
REVIEW

Cloud computing

2.1 Introduction

Cloud computing is a kind of Internet-based computing that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. There are many Cloud Service Providers (CSP) such as Google, IBM, Oracle Corporation, Amazon and Web Services. It frees a user from the concerns about the expertise in the technological infrastructure of the service. It allows the end user and small companies to make use of various computational resources like storage, software and processing capabilities provided by other companies.

The cloud services can be divided into three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Amazon, Microsoft, Google are some of the major cloud service providers. Google App Engine (GAE) is a type of PaaS provided by Google which allows web application hosting. Windows Azure, SQL Azure is some of the services offered by Microsoft providing processing and storage capabilities for large datasets. Amazon Web Services (AWS) including Simple Storage Service (S3), SQS, EC2 are cloud services provided by the Amazon. Thus convenience, on demand measured access, shared easily configurable computational resources, and self-service are some of the major characteristics of a cloud environment [2].

2.2 Cloud computing types:

There are four primary types of cloud models are:

- Public cloud.
- Private cloud (INTERNAL OR EXTERNAL).
- Community cloud (INTERNAL BY MEMBER OR EXTERNAL).
- Hybrid cloud.

2.2.1 Public cloud:

Public clouds are available to the general public or large organizations, and are owned by a third party organization that offers the cloud service. A public cloud is hosted on the internet and designed to be used by any user with an internet connection to provide a similar range of capabilities and services. Public cloud users are typically residential users and connect to the public internet through an internet service provider's network. Google, Amazon and Microsoft are examples of public cloud vendors who offer their services to the general public [3].

2.2.2 The advantages of public cloud include:

- Data availability and continuous uptime.
- 24/7 technical expertise.
- On demand scalability.
- Easy and inexpensive setup.
- No wasted resources.

2.2.3 Drawbacks of public cloud:

- Data security
- Privacy

2.2.4 Private cloud (INTERNAL OR EXTERNAL):

They are designed for exclusive use by a single organization. A private cloud may be built and managed by the organization or by external providers. It offers the highest degree of control over performance, reliability and security [4]. table 2-ghghghgh source [5]

Public vs. Private clouds

Benefit	Public	Private
Illusion of infinite resources on-demand	Yes	Unlikely
Elimination of up-front commitment by users	Yes	No
True pay-as-you-go on short-term basis	Yes	No
Economy of scale	Yes	No
Better utilization through workload multiplexing	Yes	Depends on size
Better utilization & simplified operations through virtualization	Yes	Yes

2.2.5 Community cloud:

Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized [5].

2.2.6 Hybrid cloud:

It is a combination of public cloud and private cloud. In this model a private cloud is linked to one or more external cloud services. It is more secure way to control data and applications and allows the party to access information over the internet. It enables the organization to serve its needs in the private cloud and if some occasional need occurs, it asks the public cloud for intensive computing resources [6].

2.3 Confidentiality and privacy:

Confidentiality refers to only authorized parties or systems having the ability to access protected data. The threat of data compromise increases in the cloud, due to the increased number of parties, devices and applications involved, that leads to an increase in the number of points of access. Delegating data control to the cloud, inversely leads to an increase in the risk of data compromise, as the data becomes accessible to an augmented number of parties. A number of concerns emerge regarding the issues of multi-tenancy, data permanence, application security and privacy[7].

2.4 Security Risks & Challenges are [8]:

- Conflicts with international privacy laws.
- Data ownership.
- Service guarantees.
- Securing virtual machines.
- Massive outages.
- Encryption needs & Standards.
- Storing sensitive & personal information in clouds.
- Contingency planning / disaster recovery for clouds.

2.5 Previous Studies:

2.5.1 Sanjoli and Jasmeet, “Cloud data security using authentication and encryption technique”, propose blend of two cryptographic algorithms, EAP-CHAP (Extensible Authentication Protocol- Challenge Handshake Authentication Protocol) and Rijndael Encryption Algorithm. EAP is used to provide authenticated access to the cloud environment. CHAP (method of EAP), is implemented for authentication purpose. This is then followed by encryption using Rijndael Encryption Algorithm. The complete methodology involves few steps. In the first step, Cloud Service Provider (CSP) receives an authentication request from the user. In the second step, CSP sends acknowledgement after verifying the user identity using EAP-CHAP. In the third step, once the user is authenticated, the user encrypts the data using Rijndael Encryption Algorithm and uploads the encrypted data on to the server of CSP. The data is saved in encrypted form in the server. Hence, when the user receives any encrypted data from CSP, it can be decrypted using the same key that used for encryption [9].

2.5.1.1 Result: -

They implement EAP-CHAP (Extensible Authentication Protocol-Challenge Handshake Authentication Protocol) and Rijndael Encryption Algorithm in cloud computing. will solve the authentication and authorization problems.

2.5.1.2 Open issue: -

Because of the weaknesses of Rijndael that it can be subject to standard techniques of differential and linear cryptanalysis and it is also weak to an attack called the "Square attack" that is based on the way matrix multiplication works Rijndael algorithm so that the result may not have high quality.

2.5.2 Data Confidentiality in Cloud Computing with Blowfish Algorithm

Shirole and Sanjay, "Data Confidentiality in Cloud Computing with Blowfish Algorithm", propose a system that uses encryption technique to provide a reliable and easy way to secure data for resolving security challenges. Scheduler performs encryption on plain data into cipher data followed by uploading of ciphered data on the cloud. When the data is to be retrieved from the cloud, it is obtained in plain data format and is stored on the system. This preserves data internally and hence, this builds a relationship of cooperation between operator and service provider. This model uses OTP (One-Time Password) for authentication purpose and Blowfish algorithm for encryption purpose [10].

2.5.2.1 Result: -

The function is arguably the most complex section of the algorithm and the only section that uses the Boxes. The F function accepts a 32-bit stream of data and divides the input into four equal sections. Each 8bit subdivision is transformed into a 32-bit data stream by means of their corresponding S-Box. The resulting 32 bit data is XOR'ed or added together to provide a final 32-bit value for further permutations of the Blowfish algorithm

2.5.2.2 Open issue: -

Blowfish implementations use 16 rounds of encryption, which are not susceptible to this attack. Blowfish users are encouraged by Bruce Schneier, Blowfish's creator, to use more modern and computationally efficient alternative Twofish. He is quoted in 2007 as saying "At this point, though, I'm amazed it's still being used. If people ask, I recommend Twofish instead.

2.5.3 Garima and Naveen, "Triple Security of Data in Cloud Computing", proposed a system for securing the cloud by using three algorithms: DSA (Digital Signature Algorithm), AES (Advanced Encryption Standard) and Steganography. DSA is used for authentication purpose, AES is used for encrypting the data and Steganography is used for further encryption. The work involves signing of the data in the first step. The signature is generated by first applying a hash function on the data and this gives compact form of data which is called message digest. The message digest is then signed using sender's private key. Once the message is signed, the data is encrypted along

with the signature using AES. Once encryption is completed using AES algorithm, the data is further encrypted using steganography. Steganography hides message along with another media which does attract the attention of the intruder and hence the data is protected. This complete mechanism is implemented on ASP.NET Platform and ensures to achieve authenticity, data integrity and security of data in the cloud [11].

2.5.3.1 Result: -

They implement Digital Signature Algorithm, Data Encryption Standard and Steganography to provide maximum security in cloud computing. By implementing these three algorithms, they provide authenticity, security and data integrity to that data.

2.5.3.2 Open issue: -

Time complexity is high because it is a one by one process so I try to improve the time complexity by using other suggested security algorithms.

2.5.4 Sunita and Ambrish, “Cloud Security with Encryption using Hybrid Algorithm and Secured Endpoints”, propose a hybrid algorithm for securing the cloud. In order to encrypt the message, in the first place the password is encrypted using Ceaser cipher followed by encryption using RSA substitution algorithm and then further final encryption by the mono alphabetic substitution method. Once the encryption process is over, the password is sent to the server with the plaintext user name and then the user get access to the system on successful matching. This makes the system secure and increases the speed of correction of critical issues along with

determining the root causes of vulnerability and software security assurance processes. [12]

2.5.4.1 Result: -

In this Research work, they have taken some results on the bases of algorithms which they have combined (i.e. Hybrid Algorithm) throughout their Research work in order to provide security by using PaaS (Platform as a service). For generation of encryption key, the best encryption method by combing algorithms is used.

2.5.4.2 Open issue: -

This proposed more secure techniques can be developed in order to secure cloud, for example banking application can also take advantage by expanding their transactions on public cloud rather than private cloud. But for that appropriate techniques should be developed.

2.5.5 Mamatha and Pradeep, “Use of Digital Signature with Diffie Hellman Key Exchange and Hybrid Cryptographic algorithm to Enhance Data Security in Cloud Computing”, propose a hybrid algorithm for securing the cloud, using three ways of protection scheme. Firstly, to generate keys for key exchange step, Diffie Hellman algorithm is used. Then digital signature is used for authentication, there after user’s data file is encrypted or decrypted using hybrid encryption algorithm. With hybrid algorithm data will be uploaded into cloud server by double encryption. Initially data will be encrypted using AES algorithm and again re-encryption will be done by

3DES and similar lily data will be downloaded from the cloud server by decrypting the file as exactly reverse of encryption process [13].

2.5.5.1 Result: -

They implement combined concept of AES and 3DES to reduce the possibility of an algebraic attack on the hybrid model.

2.5.5.2 Open issue: -

To solve the problem and protect the key from the attackers, an algorithm is suggested to design a function works as a shield covering the key to disable to be hacked and no one can get the encrypted text even he/she breaks the key because of the existence of the shield.

	Paper Name	Date	Publisher/ author	Techniques
1	Cloud Data Security using Authentication and Encryption Technique	2013	Global Journal of Computer Science and Technology Cloud and Distributed / SanjoliSingla and Jasmeet Singh	EAP-CHAP(Extensible Authentication Protocol- Challenge Handshake Authentication Protocol) and Rijndael Encryption Algorithm.
2	Data Confidentiality in Cloud Computing with Blowfish Algorithm	2014	International journal of Emerging Trends in Science and Technology / Dr. Sanjay Thakur	Blowfish encryption algorithm
3	Triple Security of Data in Cloud Computing	2014	International Journal of Computer Science and Information Technologies / GarimaSaini	DSA (Digital Signature Algorithm), AES (Advanced Encryption Standard) and Steganography
4	Cloud Security with Encryption using Hybrid Algorithm and Secured Endpoints	2012	International Journal of Computer Science and Information Technologies/ Sunita Rani, AmbrishGangal	Hybrid Algorithm
5	Use of Digital Signature with Diffie Hellman Key Exchange and Hybrid Cryptographic algorithm to Enhance Data Security in Cloud Computing	2015	International Journal of Scientific and Research Publications / Mrs.Mamatha, Mr.PradeepKanchan	hybrid cryptographic algorithm (Advanced Encryption Standard (AES) and Data Encryption Standard (DES)) , digital signature and Diffie Hellman key exchange.

Table 2.1 Summary of related works

CHAPTER III
METHODOLOGY, TOOLS AND
TECHNIQUES

RESEARCH FRAME WORK

3.1 Algorithms:

The algorithm depends on the Key, the Key Generator of the ID, and each ID has a different value than the other, leading to algorithm strength, because each user has a different ID than the other.

3.2 Generate Key function:

It generates six keys as follows:

- Receive user ID.
- Convert it to MD5.
- Generate Keys.

It receives an ID and converts it to MD5, which results in extracting a 32-digit code and store in a variable Called key. Then it is divided it by using the split function and store it in a variable called SplitKey (SK). It is stored in the form of a matrix, then the numbers and letters are separated from MD5 using foreach function, and then extract the numbers only from MD5 using ord function.

3.3 Encrypt function:

After generating the six keys, the text is received to be encrypted, and then put it in the form of a matrix through the split function, and convert the text into numbers by using ord function (the letter with which number in the ASCII table), and then a test of Capital, Small, Number is performed, because in each case we call a specific function.

Before calling the function that searches for the letter or character in the ASCII table, a key is added.

If it is the first character, the first key is added. If it is the second character, the second key is added. If it is the third character, the third key is added and so on.

The idea is that every six digits are encoded with six keys.

After adding the character of the encryption value, for example if the character value $x = 120$, and the value of the encryption (Key) = 9.

Then $120 + 9 = 129$

129 out of range in the ASCII Table because the last value in ASCII Table is 127, to solve this problem we send it to the Ensmall function.

3.4 Ensmall Function:

Purpose of Ensmall Function to Ensure that the number is within range, for example the character Small between 97 - 122, if within the range the function returns the same value, if it is out of range we subtract 26 to enter the number range, and then return the value to the encrypt function.

Then we call the variable Encrypt Text, initially a null value, and the empty variable we add the value of encryption (Key) and then we convert the encrypted number to a character as in the previous example:

Character x

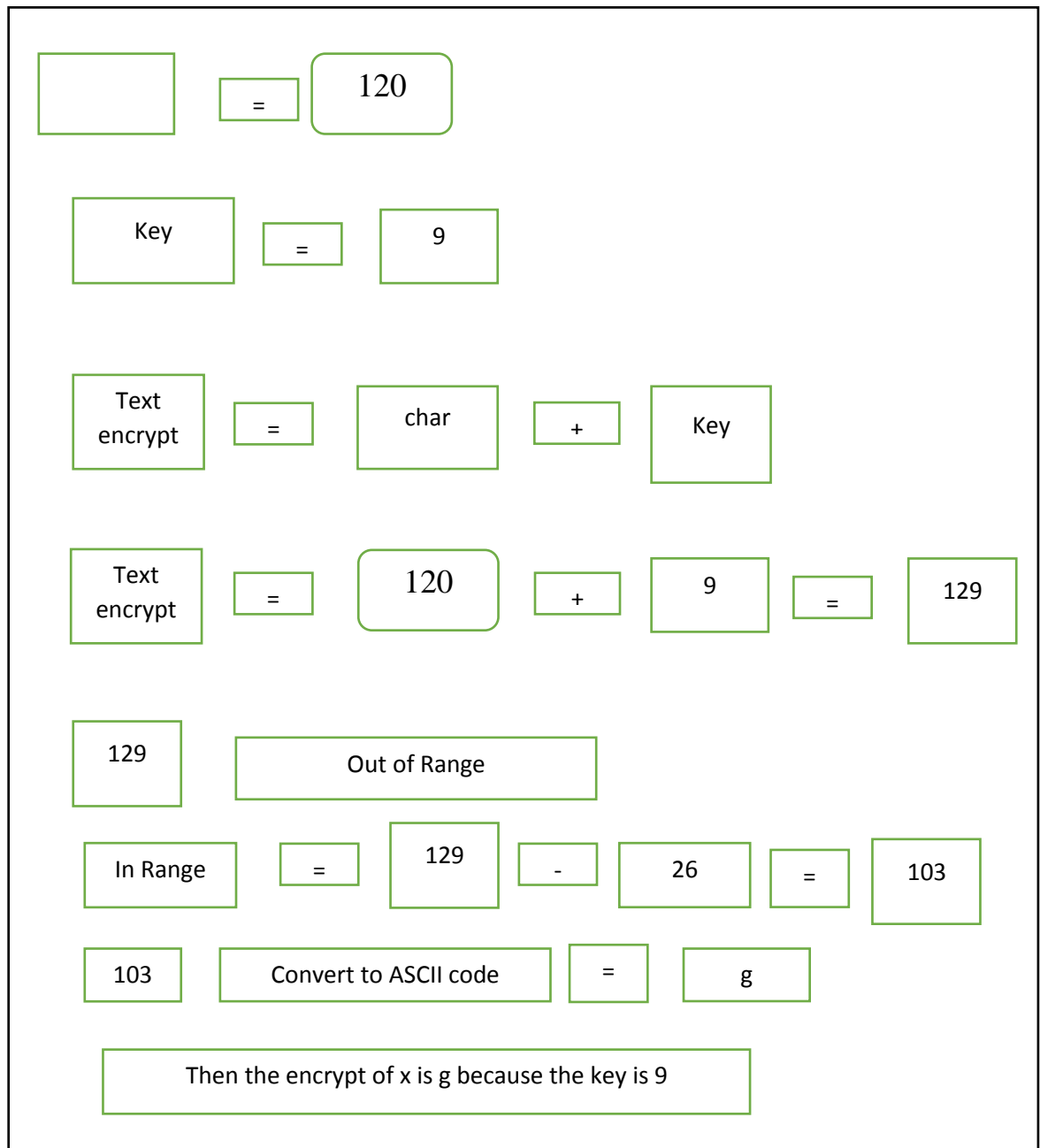
Value of $x = 120$

Key = 9

then $120 + 9 = 129$

Subtract the value from the Range (26)

$$129 - 26 = 103$$



Example of encrypt Text

Thus (x) is (g) because the key equals 9, each time the key is different from the other.

For example, if 6 are another value, the greater the value of the key and so on...

After the encryption process is finished, we return the collected characters in matrix format.

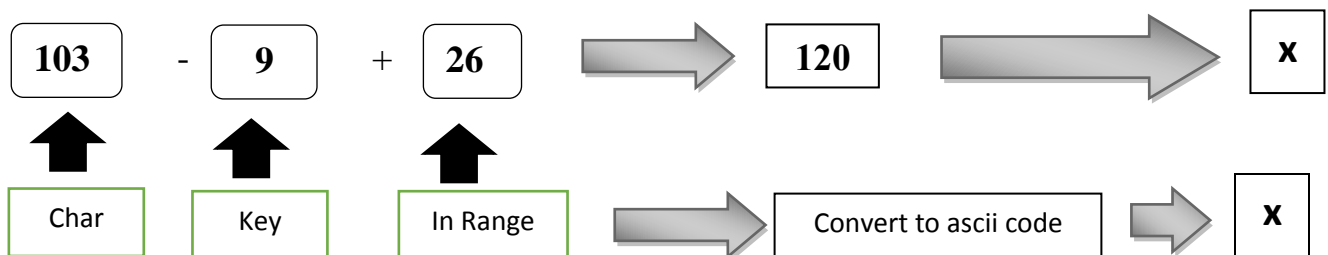
3.5 Decryption Function:

Is the reverse process of encryption we do the same steps unlike the tags?

- Instead of adding the value of the encryption (Key) we subtract the value of the encryption.
- Check the number within the range instead of the subtraction. We are adding 26.
- ❖ For example, an x character decodes the character g as in the previous example, to return the old value

It was a value of g is 103

subtraction the key 9 and add 26



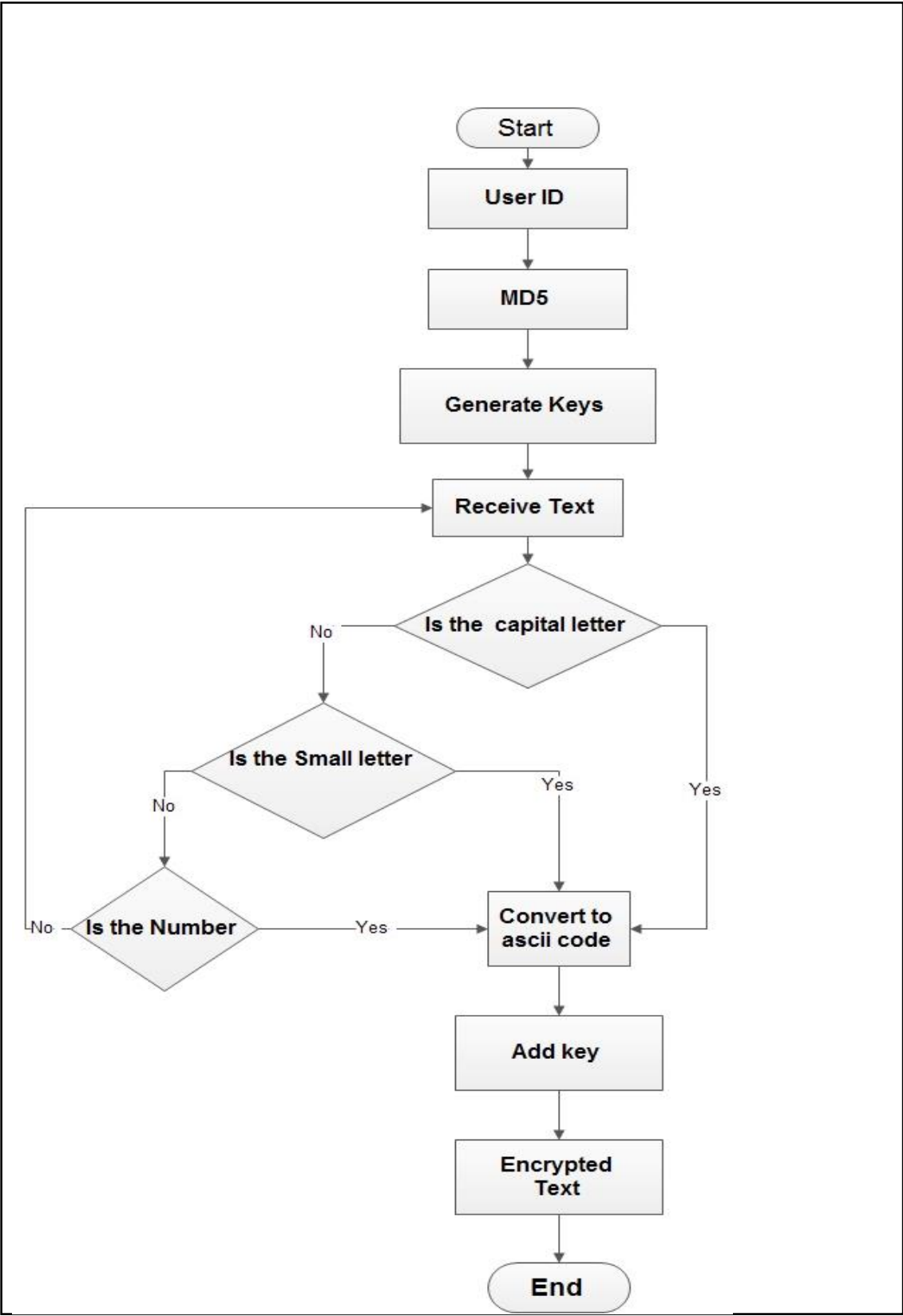
3.6 PHP:

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus, Lerdorf in 1994, the PHP reference

implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor [14].

3.7 Message Digest 5 (MD5):

MD5 is a message digest algorithm developed by Ron Rivest at MIT. It is basically a secure version of his previous algorithm, MD4 which is a little faster than MD5. This has been the most widely used secure hash algorithm particularly in Internet-standard message authentication. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest of the input. This is mainly 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 [15].



Encryption Flow Chart

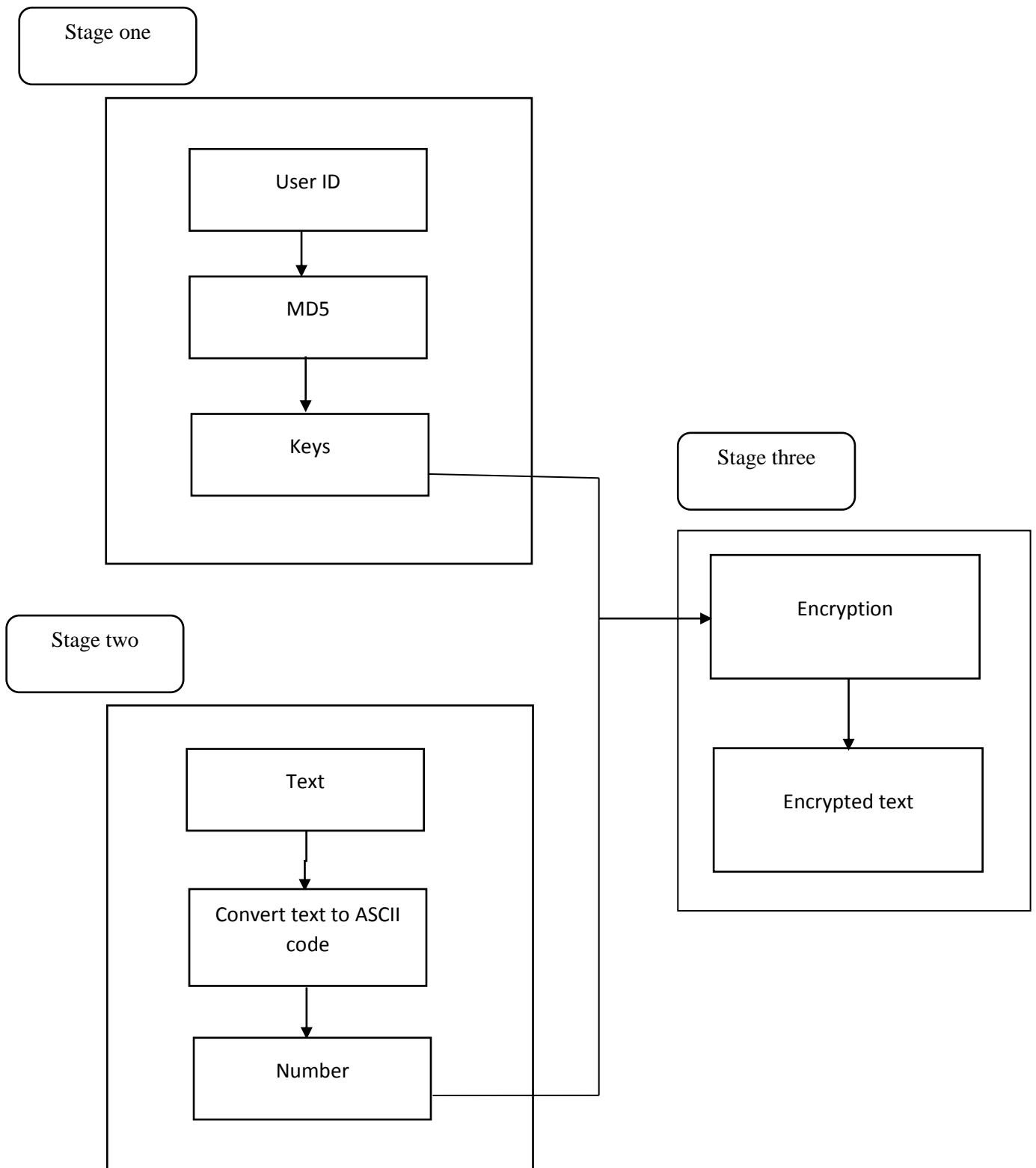


Figure (3.1) Framework for encrypt text

CHAPTER IV
IMPLEMENTATION

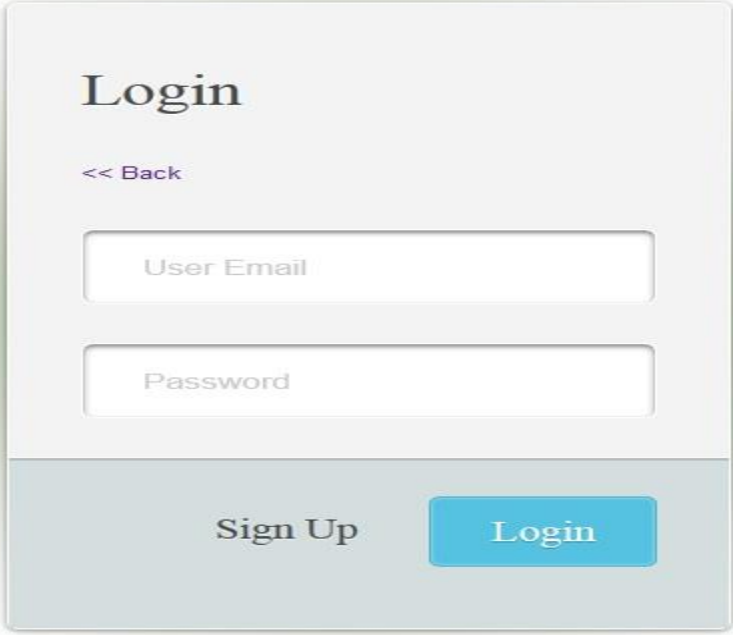
4.1 Introduction

In this chapter we present the Implementation of Encryption Algorithm and show the system screens.

4.2 System Screens:

Figure 4.1 show login form that uses by user to enter the system. Which enter User Email and Password and press on Login Button to login the system.

After the Press Login Button the system will check whether the information had entered that belonging to authorize user.

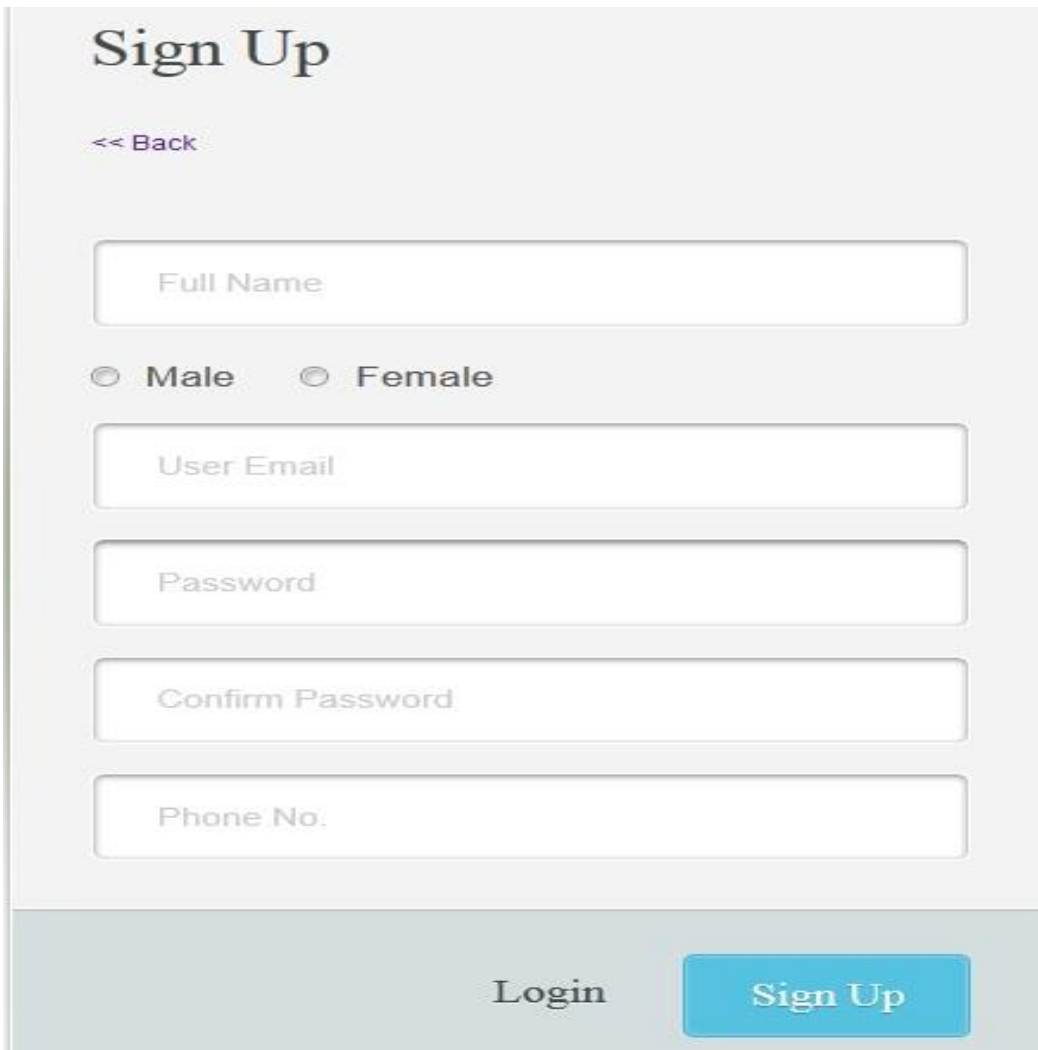


The image shows a login form with a light gray background. At the top, the word "Login" is written in a serif font. Below it is a link that says "<< Back". There are two input fields: the first is labeled "User Email" and the second is labeled "Password". At the bottom of the form, there are two buttons: "Sign Up" and "Login". The "Login" button is highlighted in a blue color.

Figure (4.1) login form

Figure 4.2 show Sign up Form that uses by user to register the system. Which enter Full Name, select gender (Male, Female), user Email,

Password, confirm Password and Phone Number, and press on sign Up Button to register the system, After the Press sign Up Button the system will saved information in the Database.



The image shows a 'Sign Up' form with a light gray background. At the top left, the title 'Sign Up' is displayed in a dark serif font. Below the title is a '<< Back' link. The form contains several input fields: 'Full Name', 'User Email', 'Password', 'Confirm Password', and 'Phone No.'. Between the 'Full Name' and 'User Email' fields, there are two radio buttons labeled 'Male' and 'Female'. At the bottom of the form, there are two buttons: a 'Login' button and a blue 'Sign Up' button.

Figure (4.2) Sign Up Form

Figure 4.3 show attachment Form that uses by user, then press on Browse Button to upload file in the system.

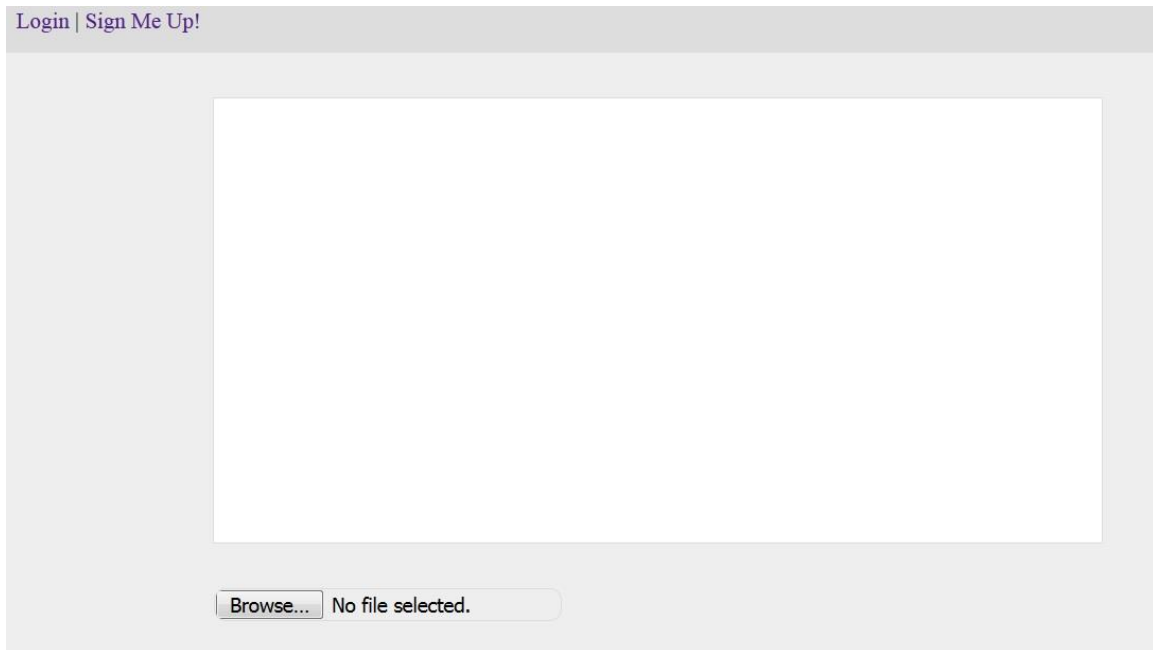


Figure (4.3) attachment Form

Figure 4.4 show Encrypt button that uses by user, after upload file, press Encrypt Button to encrypt file in the system.

When to Decrypt File, press Decrypt Button to Decrypt file in the system.

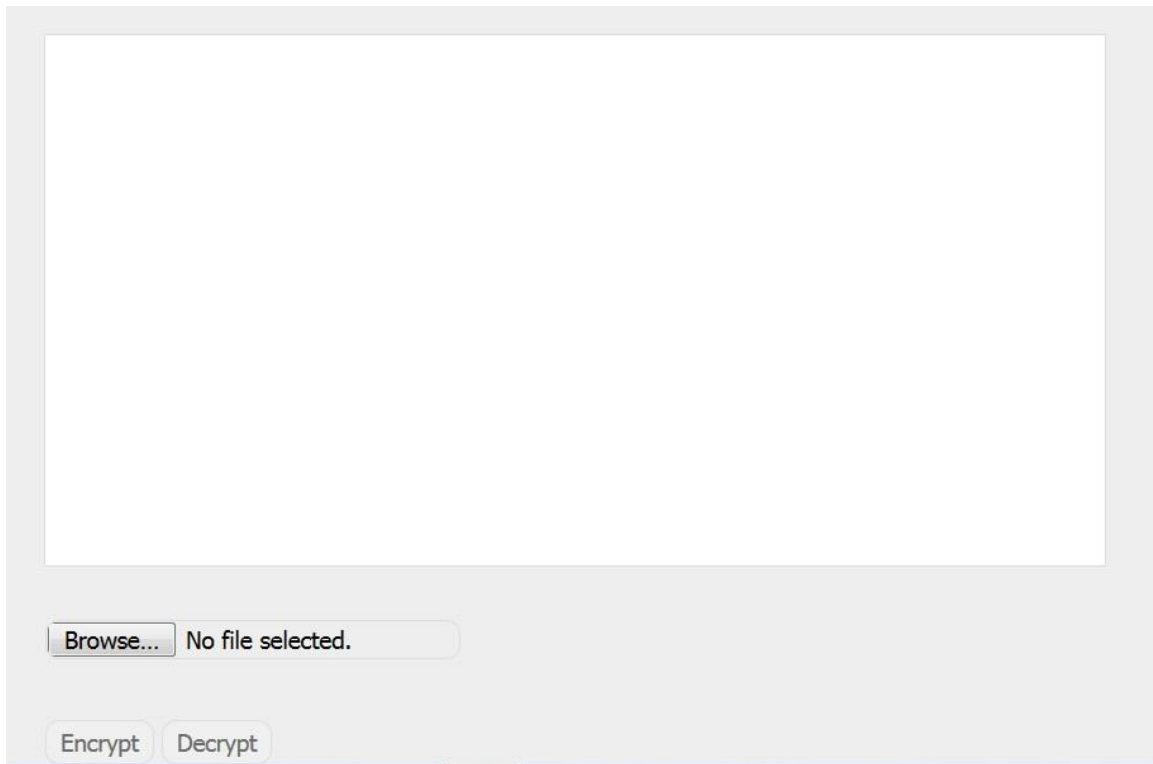


Figure (4.4) Encrypt or decrypt Button.

Figure 4.5 show Encrypted uploaded files in the system.

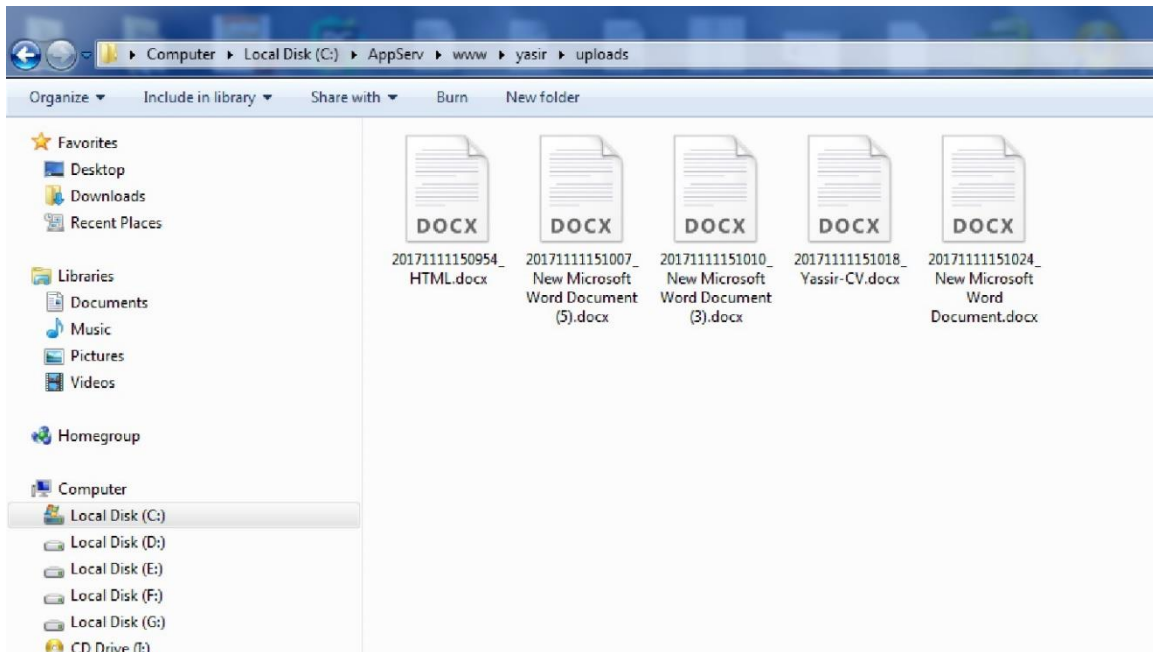


Figure (4.5) upload files

4.3 RESULTS: -

From the implementation of the proposed model, the results are protecting the original text form the unauthorized users, generate unique key from one user to another, generate key is depending on user ID and low cost to implement the system.

The next table shows the comparison the proposed model to models which discussed as related works in chapter II.

Algorithms	key size	Block Size	Round	Structure	Flexible	Features
DES	64 bits	64 bits	16	Feistel	No	Not structure, Enough
3DES	112 or 118 bits	64 bits	48	Feistel	Yes	Adequate security
AES	128,192,256 bits	128 bits	10,12,14	Substitution, Permutation	Yes	replacement for DES, Excellent security
BLOW FISH	32-448 bits	64 bits	16	Feistel	Yes	Excellent security

From this table we can conclude the blow fish algorithm is more secure to compare other symmetric key algorithms, and produce best result for less processing time and rounds. To increase the key size of blowfish algorithm 128 to 448, it gives more privacy to the messages and provides high end data security when transmitting over any unsafe medium. In above table show the blowfish algorithm is providing excellent security to compare symmetric algorithms.

In Research uses group of algorithms to provide more security in the system.

CHAPTER V

CONCLUSION AND

RECOMMENDATIONS

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion:

Cloud Computing has been envisioned as the next generation architecture of IT Enterprise. In cloud computing application software and databases are moving to the centralized large data centers. This mechanism brings about many new challenges, which have not been well understood. Security and privacy concerns, however, are among the top concerns standing in the way of wider adoption of cloud. In cloud computing the main concern is to provide the security to end user to protect files or data from unauthorized user. Security is the main intention of any technology through which unauthorized intruder can't access your file or data in cloud. We have to design an Algorithm that can help to encrypt and decrypt the file at the user side that provide security to data at rest as well as while moving. In this research, we have used the md5 Encryption Algorithm to generate key, the algorithm depends on the key, and the key is generated from the user ID, and each user ID has a different value than the other this Lead to a powerful algorithm, because each user has a different ID than the other.

5.2 Recommendations:

In this research it is highly recommended for future researchers to develop specific encryption algorithms for other languages like Arabic Language. furthermore, it is also required to consider the symbols during the encryption task.

REFERENCE:

- [1] Hamdaqa, M., &Tahvildari, L. (2012). Cloud computing uncovered: a research landscape. *Advances in Computers*, 86(41), 43-84.
- [2] Maral, V., Kale, S., Balharpure, K., Bhakkad, S., &Hendre, P. (2016). Homomorphic Encryption for Secure Data Mining in Cloud.*International Journal of Engineering Science*, 4533.
- [3] Zhang, Q., Cheng, L., &Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges.*Journal of internet services and applications*, 1(1), 7-18.
- [4] Singla, S., & Singh, J. (2013). Cloud data security using authentication and encryption technique.*Global Journal of Computer Science and Technology*, 13(3).
- [5] Subhash, S. B., & Thakur, D. S. (2014). Data Confidentiality in Cloud Computing with Blowfish Algorithm.*International Journal of Emerging Trends in Science and Technology*, 1(01).
- [6] Garima and Naveen. “Triple Security of Data in Cloud Computing”, *International Journal of Computer Science and Information Technologies* 2014.
- [7] Rani, S., &Gangal, A. (2012). Cloud security with encryption using hybrid algorithm and secured endpoints. *International journal of computer science and information technologies*, 3, 4302-4304.
- [8] Mamatha, M., &Kanchan, M. P. (2015). Use of Digital Signature with Diffie Hellman Key Exchange and Hybrid Cryptographic algorithm to Enhance Data Security in Cloud Computing.*International Journal of Scientific and Research Publications*, 5(6).

- [9] Lerdorf, R., Tatroe, K., &MacIntyre, P. (2006). Programming Php. " O'Reilly Media, Inc."
- [10] Rivest, R. (1992). The MD5 message-digest algorithm.
- [11] Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: a critical review. *International Journal of Computer Network and Information Security*, 6(3), 20.
- [12] Bora, U. J., & Ahmed, M. (2013). E-learning using cloud computing. *International Journal of Science and Modern Engineering*, 1(2), 9-12.
- [13] Zissis, D., &Lekkas, D. (2012). Addressing cloud computing security issues.*Future Generation computer systems*, 28(3), 583-592.
- [14] Jadeja, Y., &Modi, K. (2012, March). Cloud computing-concepts, architecture and challenges.In *Computing, Electronics and Electrical Technologies (ICCEET)*, 2012 International Conference on (pp. 877-880).IEEE.
- [15] Xiao, Z., & Xiao, Y. (2013). Security and privacy in cloud computing. *IEEE Communications Surveys & Tutorials*, 15(2), 843-859.

Appendices

```
1 <?php
2 if(!isset($_COOKIE["UserId"])){
3     echo "You don't have permission to do this operation, Please login first.";
4     exit;
5 }
6 $id = $_COOKIE["UserId"];
7 $txt = $_POST['Txt'];
8 $Op = $_POST['Operation'];
9
10 if($Op == 1){
11     echo Encrypt($txt,$id)."<br>";
12 }
13 if($Op == 2){
14     echo Decrypt($txt,$id)."<br>";
15 }
16
17
18 function Decrypt($Text,$Key){
19     $Key = GenerateKey($Key);
20     $Text = str_split($Text);
21     $L = 0;
22     $DecryptText="";
23     foreach($Text As $Txt){
24         $c = ord($Txt);
25         if($c >= 65 and $c <= 90)
26         {
27             $c = DeCapital($c - $Key[$L]);
28         }
29         elseif($c >= 97 and $c <= 122)
30         {
31             $c = DeSmall($c - $Key[$L]);
32         }
33         elseif($c >= 48 and $c <= 57)
```

```

44
45 function Encrypt($Text,$Key){
46     $Key = GenerateKey($Key);
47     $Text = str_split($Text);
48     $L = 0;
49     $EncryptText="";
50     foreach($Text As $Txt){
51         $c = ord($Txt);
52         if($c >= 65 and $c <= 90)
53         {
54             $c = EnCapital($c + $Key[$L]);
55         }
56         elseif($c >= 97 and $c <= 122)
57         {
58             $c = EnSmall($c + $Key[$L]);
59         }
60         elseif($c >= 48 and $c <= 57)
61         {
62             $c = EnNumber($c + $Key[$L]);
63         }
64         if($L >= 5) $L = 0; else $L = $L + 1;
65         $EncryptText = $EncryptText.chr($c);
66     }
67     return $EncryptText;
68 }
69
70 function GenerateKey($ID){
71     $key = md5($ID);
72     $splittedKey = str_split($key);
73     $sK="";

```

```

83 function EnCapital($n){
84     if($n >= 65 and $n <= 90){
85         return $n;
86     }
87     else {
88         $n = $n - 26;
89         EnCapital($n);
90     }
91     return $n;
92 }
93 function EnSmall($n){
94     if($n >= 97 and $n <= 122){
95         return $n;
96     }
97     else {
98         $n = $n - 26;
99         EnSmall($n);
100    }
101    return $n;
102 }
103 function EnNumber($n){
104     if($n >= 48 and $n <= 57){
105         return $n;
106     }
107     else {
108         $n = $n - 10;
109         EnNumber($n);
110     }
111     return $n;
112 }
113

```

```

113
114 function DeCapital($n){
115     if($n >= 65 and $n <= 90){
116         return $n;
117     }
118     else {
119         $n = $n + 26;
120         EnCapital($n);
121     }
122     return $n;
123 }
124 function DeSmall($n){
125     if($n >= 97 and $n <= 122){
126         return $n;
127     }
128     else {
129         $n = $n + 26;
130         EnSmall($n);
131     }
132     return $n;
133 }
134 function DeNumber($n){
135     if($n >= 48 and $n <= 57){
136         return $n;
137     }
138     else {
139         $n = $n + 10;
140         EnNumber($n);
141     }
142     return $n;
143 }
144 ?>

```

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>Sign-Up</title>
5 </head>
6 <body id="body-color">
7 <div id="Sign-Up">
8 <fieldset style="width:30%"><legend>Registration Form</legend>
9 <table border="0">
10 <tr>
11 <form method="POST" action="connectivity-sign-up.php">
12 <td>Name</td><td> <input type="text" name="name"></td>
13 </tr>
14 <tr>
15 <td>Email</td><td> <input type="text" name="email"></td>
16 </tr>
17 <tr>
18 <td>UserName</td><td> <input type="text" name="user"></td>
19 </tr>
20 <tr>
21 <td>Password</td><td> <input type="password" name="pass"></td>
22 </tr>
23 <tr>
24 <td>Confirm Password </td><td><input type="password" name="cpass"></td>
25 </tr>
26 <tr>
27 <td><input id="button" type="submit" name="submit" value="Sign-Up"></td>
28 </tr>
29 </form>
30 </table>
31 </fieldset>
32 </div>

```

```

<div class="form-container">
<?php echo $output; ?>
<form class="form" action="process.php" method="post" id="register-form" novalidate="novalidate">
<fieldset>
<ol>
<li class="form-row text-input-row">
<label>Full Name</label>
<input type="text" name="name" value="<? echo $name; ?>" class="text-input required" title="
</li>
<li class="form-row select-row">
<label>Nationality</label>
<select name="nat" class="select-input">
<option value=""> ----- Select Nationality ----- </option>
<?php
include'include/connection.php';
$sql = "SELECT * FROM country";
$result = mysqli_query($con,$sql);
while($row = mysqli_fetch_array($result))
{
echo '<option value='.$row['country_id'].'>';
echo $row['country_code'].'&nbsp;&nbsp;&nbsp;'.$row['country_name'].'</option>';
}
?>
</select>
</li>
<li class="form-row select-row">
<label>Gender</label>
<select name="gender" class="select-input">
<option value=""> ----- Select Gender ----- </option>
<option value="0">Male</option>
<option value="1">Female</option>
</select>
</li>

</ol>
<li class="form-row select-row">
<label>Gender</label>
<select name="gender" class="select-input">
<option value=""> ----- Select Gender ----- </option>
<option value="0">Male</option>
<option value="1">Female</option>
</select>
</li>
<li class="form-row text-input-row">
<label>Email</label>
<input type="text" name="email" value="" class="text-input required email" title="" />
</li>
<li class="form-row text-input-row">
<label>Password</label>
<input type="password" name="password" value="" class="text-input required" title="" />
</li>
<li class="form-row text-input-row">
<label>Confirm Pass.</label>
<input type="password" name="repass" value="" class="text-input required" title="" />
</li>
<li class="form-row text-input-row">
<label>Phone</label>
<input type="text" name="phone" value="" class="text-input required" title="" />
</li>
<li class="form-row text-input-row">
<label>Other Phone</label>
<input type="text" name="otherphone" value="" class="text-input" title="" />
</li>
<li class="form-row text-input-row">
<label>Additional Phone</label>
<input type="text" name="addphone" value="" class="text-input" title="" />
...

```

```

64     <input type="text" name="otherphone" value="" class="text-input" title="" />
65 </li>
66 <li class="form-row text-input-row">
67     <label>Additional Phone</label>
68     <input type="text" name="addphone" value="" class="text-input" title="" />
69 </li>
70 <li class="form-row text-input-row">
71     <label>Fax</label>
72     <input type="text" name="fax" value="" class="text-input" title="" />
73 </li>
74 <li class="form-row text-input-row">
75     <label> Date of Birth</label>
76     <input type="text" name="DOB" id="datepicker" value="" class="text-input" title="" />
77 </li>
78 <li class="form-row hidden-row">
79     <input type="hidden" name="form" value="signup" />
80     <input type="hidden" name="page" value="Sign Up" />
81     <input type="hidden" name="p" value="<?php echo $p; ?>" />
82 </li>
83 <li>
84     <div style="margin-left: 100px;">By clicking Sign Up, you agree to our <a style="text-decorat
85 </li>
86 <li class="button-row">
87     <input type="submit" value="Sign Up" name="submit" class="btn-submit" />
88 </li>
89 </ol>
90 <input type="hidden" name="v_error" id="v-error" value="Required" />
91 <input type="hidden" name="v_email" id="v-email" value="Enter a valid email" />
92 </fieldset>
93 </form>
94 <div class="response"></div>
95 </div><br /> <br />
96 !-- End Form -->

```

```

1 <!DOCTYPE html>
2 <html>
3 <style>
4 /* Full-width input fields */
5 input[type=text], input[type=password] {
6     width: 100%;
7     padding: 12px 20px;
8     margin: 8px 0;
9     display: inline-block;
10    border: 1px solid #ccc;
11    box-sizing: border-box;
12 }
13
14 /* Set a style for all buttons */
15 button {
16     background-color: #4CAF50;
17     color: white;
18     padding: 14px 20px;
19     margin: 8px 0;
20     border: none;
21     cursor: pointer;
22     width: 100%;
23 }
24
25 button:hover {
26     opacity: 0.8;
27 }
28
29 /* Extra styles for the cancel button */
30 .cancelbtn {
31     width: auto;
32     padding: 10px 18px;
33     background-color: #f44336;

```

```

40     position: relative;
41 }
42
43 img.avatar {
44     width: 40%;
45     border-radius: 50%;
46 }
47
48 .container {
49     padding: 16px;
50 }
51
52 span.psw {
53     float: right;
54     padding-top: 16px;
55 }
56
57 /* The Modal (background) */
58 .modal {
59     display: none; /* Hidden by default */
60     position: fixed; /* Stay in place */
61     z-index: 1; /* Sit on top */
62     left: 0;
63     top: 0;
64     width: 100%; /* Full width */
65     height: 100%; /* Full height */
66     overflow: auto; /* Enable scroll if needed */
67     background-color: rgb(0,0,0); /* Fallback color */
68     background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
69     padding-top: 60px;
70 }
71
72
73
74
75
76     border: 1px solid #000;
77     width: 80%; /* Could be more or less, depending on screen size */
78 }
79
80 /* The Close Button (x) */
81 .close {
82     position: absolute;
83     right: 25px;
84     top: 0;
85     color: #000;
86     font-size: 35px;
87     font-weight: bold;
88 }
89
90 .close:hover,
91 .close:focus {
92     color: red;
93     cursor: pointer;
94 }
95
96 /* Add Zoom Animation */
97 .animate {
98     -webkit-animation: animatezoom 0.6s;
99     animation: animatezoom 0.6s
100 }
101
102 @-webkit-keyframes animatezoom {
103     from {-webkit-transform: scale(0)}
104     to {-webkit-transform: scale(1)}
105 }
106
107 @keyframes animatezoom {
108     from {transform: scale(0)}

```

```

121 }
122 </style>
123 <body>
124
125 <h2>Modal Login Form</h2>
126
127 <button onclick="document.getElementById('id01').style.display='block'" style="width:auto;">Login</button>
128
129 <div id="id01" class="modal">
130
131 <form class="modal-content animate" action="/action_page.php">
132 <div class="imgcontainer">
133 <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">
134 </div>
135
136 <div class="container">
137 <label><b>Username</b></label>
138 <input type="text" placeholder="Enter Username" name="uname" required>
139
140 <label><b>Password</b></label>
141 <input type="password" placeholder="Enter Password" name="psw" required>
142
143 <button type="submit">Login</button>
144 <input type="checkbox" checked="checked" Remember me
145 </div>
146
147 <div class="container" style="background-color:#f1f1f1">
148 <button type="button" onclick="document.getElementById('id01').style.display='none'" class="cancel">
149 <span class="psw">Forgot <a href="#">password?</a></span>
150 </div>
151 </form>
152 </div>
...

```

```

139
140 <label><b>Password</b></label>
141 <input type="password" placeholder="Enter Password" name="psw"
142
143 <button type="submit">Login</button>
144 <input type="checkbox" checked="checked" Remember me
145 </div>
146
147 <div class="container" style="background-color:#f1f1f1">
148 <button type="button" onclick="document.getElementById('id01');
149 <span class="psw">Forgot <a href="#">password?</a></span>
150 </div>
151 </form>
152 </div>
153
154 <script>
155 // Get the modal
156 var modal = document.getElementById('id01');
157
158 // When the user clicks anywhere outside of the modal, close it
159 window.onclick = function(event) {
160     if (event.target == modal) {
161         modal.style.display = "none";
162     }
163 }
164 </script>
165
166 </body>
167 </html>

```