

# Implementation of Hybrid Encryption Algorithm

Monu Agrawal<sup>1</sup>, Riwaz Udas<sup>2</sup>, Pranav Kumar Jha<sup>3</sup>

<sup>1,2,3</sup>Student, VIT University

## Abstract

The vulnerabilities of using a single encryption system in messaging systems has proven to become increasingly problematic in recent days, this paper provides a new form of encryption to be used in such systems. The proposed system is a new combination of two very prevalent encryption algorithms namely the Vigenere encryption algorithm and the Base-64 encryption (also known as B64 encryption). The combined encryption algorithm is made by using the Base-64 encryption followed by the Vigenere algorithm. Each message sent by the user is secured after being greatly enhanced through this encryption technique.

## Keywords

Vigenere encryption algorithm; Base-64 encryption algorithm, RSA algorithm, DES algorithm, Plain Text, Cipher Text, Polyalphabetic substitution, Hybrid encryption algorithm, Encryption key, Decryption Key, Key Wrapping, Private Key.

## 1. Introduction

With the development in the processing capabilities of computers in recent years and the widespread use of the Internet, the transfer of data through the use of Internet has increased drastically. The lives of people have become thoroughly spread out into the Web. But due to the increase in processing powers of computer, the previously used highly secured encryption systems have been rendered useless. The need for a new safer, more difficult to hack security algorithm has become imminent, in order to secure the privacy of all those who use Internet to share data. This series of requirements has become an active research topic in the field of IT as well as network security.

A proper encryption algorithm is the most basic software-based security technique being used in modern messaging systems. The policies used by modern companies that sell data and message histories to other third party has become a news topic all around the world, hence a proper encryption is a must in all applications used for this purpose.

At present the most popular encryption technology used is DES encryption, RSA encryption algorithm, Base-64 and Vigenere encryption algorithm. But even these encryption systems have become vulnerable due to the use of large computer farms in order to hack them. In regard to this issue the use of a hybrid encryption techniques consisting of different encryption algorithms has been used, we have first defined a Base-64 encryption algorithm in this paper, followed a Vigenere encryption, together forming a new hybrid encryption algorithm. The hybrid encryption algorithm has now greatly reduced the risk of possible data leaks from messaging systems in the future.

## 2. Problem Statement

The increased number of data theft in various messaging apps in recent years has brought about the need for a strong unbreakable encryption of our data to ensure our privacy. In addition to the message sent over the internet, our data stored in servers needs to be enhanced greatly. Hence a end to end encryption that can only be decrypted by the involved parties is necessary to be implemented. The key idea of this project is to provide users to a encryption system called the hybrid encryption system, which can provide two layers of security instead of the traditional one, to protect user data from being breached. The application also provides an illustration of the use of the hybrid encryption system through the implementation of a messaging system and it also displays the capability of the encryption algorithm in order to make the user data to be as secure as possible.

## 3. Proposed Methodology

In this method we plan to take a string and encrypt it first with Base-64 encryption.

Post encryption the encrypted message along with private key is passed to the Vigenere encryption function. The private key length will be extended to the length of the Base-64 encrypted message filling the remaining characters with characters from the original private key and this function to extend the key can also be changed separately to wrap messages according to the user’s preferences. After the new creation of a new private key the Vigenere encryption on the message starts along with the newly generated private key. The string being returned after that is result of the hybrid encryption method proposed after making use of both the Vigenere encryption and Base-64 encryption method.

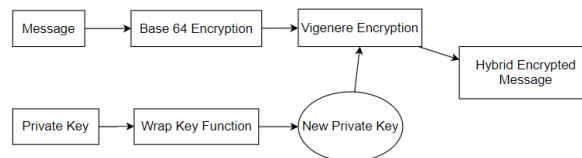


Diagram 1: System Architecture of Hybrid Encryption Algorithm

For Decryption the process is completely reverse of what is done for encryption.

The hybrid encrypted message is passed to the decryption function along with the same private key that was used to encrypt it. The private key is wrapped or extended along with hybrid encrypted message. The hybrid encrypted message is then decrypted with the help of new extended private key to bring it back to Base-64 encrypted code. The Base-64 decryption process is done which is the reverse process of Base-64 encryption in order to obtain the decrypted message.

The string post decryption of Vigenere and post Base-64 encryption indicates that the function responsible for extension of the key is working well.

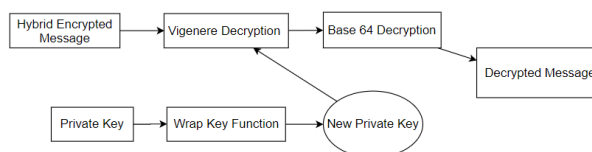


Diagram 2: System Architecture of Hybrid Encryption Algorithm

## 4. Algorithm-Steps and Explanation

The proposed hybrid encryption algorithm involves the following steps:

## **I. Encryption Process**

### ***a) Base 64 Encryption***

- i. The message is passed to the function which encrypts the message by following the traditional B64 encoding process where the bits are shifted and manipulated after the conversion of the characters from their ASCII to binary values.
- ii. Padding is also done before grouping the binary values of the messages before returning the B64 encrypted string.

### ***b) Vigenere Encryption***

- i. The encrypted Base-64 message is passed onto the function along with private key.
- ii. The private key is extended to the length of the Base-64 message where the remaining characters for the private key are taken from the original private key giving rise to a new extended private key
- iii. The Vigenere encryption process starts around the Base64 encrypted message along with the newly extended private key and thus returning a new string that has undergone the hybrid encryption algorithm.

## **II. Decryption process**

### ***a) Vigenere Decryption***

- i. The hybrid encrypted message is received along with the private key.
- ii. The private key is extended to the length of the Hybrid encrypted message where the remaining characters for the private key are taken from the original private key giving rise to a new extended private key.
- iii. The Vigenere decryption is then carried out with the help of the newly extended private key and a B64 encrypted string is the result.

### ***b) Base-64 Decryption***

- i. The B64 encrypted message is received which then converted into binary values from ASCII values of its character.
- ii. The binary values are then grouped in a way completely reverse to the encryption process and then accordingly padded.
- iii. The binary values are converted into the corresponding ASCII and into the characters framing the decrypted string which is the decrypted message.

## **5. Design of Hybrid Encryption Algorithm**

### ***A. Base-64 Encryption Algorithm***

Base-64 encoding algorithm encodes binary data using only printable text characters. A normal ASCII text character is made up of 8-bit binary characters. Base-64 encryption converts the 8-bit text character into a new 6-bit character format. For a 3-byte character, the algorithm first converts it into a 24-bit binary character string. The new 24-bit binary character is then divided into 4 segments of 6 bits each. It is then compared to a table of B64 character table the algorithm uses to convert the new 4 6-bit segments into new 4 text character long string. For example, the string “NETSEC” is converted to “TkVUU0VD” after processing it through the algorithm.

decimal	0	1	2	3	4	5	6	7	8	9	10
character	A	B	C	D	E	F	G	H	I	J	K
decimal	11	12	13	14	15	16	17	18	19	20	21
character	L	M	N	O	P	Q	R	S	T	U	V
decimal	22	23	24	25	26	27	28	29	30	31	32
character	W	X	Y	Z	a	b	c	d	e	f	g
decimal	33	34	35	36	37	38	39	40	41	42	43
character	h	i	j	k	l	m	n	o	p	q	r
decimal	44	45	46	47	48	49	50	51	52	53	54
character	s	t	u	v	w	x	y	z	0	1	2
decimal	55	56	57	58	59	60	61	62	63		
character	3	4	5	6	7	8	9	+	/		

Table 1: Base64 algorithm character set

As the number of characters present in the ciphertext is longer than the plain text. We can come to a conclusion that the plaintext has been broken down and converted into characters with 6-bit binary characters. In scenarios where the 8-bit ASCII value is not perfectly divisible by six, new bits are added to the plain text in order to make it divisible, zeros are added to the end, through the process of padding.

**B. Vigenere Encryption Algorithm**

Vigenere Cipher is an encrypting algorithm, it uses a form of polyalphabetic substitution to encrypt a message. This mode of substitution involves multiple substitution of the characters in the message. The substitution is performed by comparing the message and the encryption key in a Vigenere’s table and the intersecting point between the two characters is taken as the new cipher text.

Taking an example for the plaintext “NETSEC” and the encryption key “KEY”, we first extend the encryption key to match the length of the plaintext using the process of text wrapping. The new encryption key will be “KEYKEY”. After this is done each character in both the plain text and the key is compared in the character set table for Vigenere algorithm. For ‘N’ and ‘K’ the new encrypted character will be ‘X’ similarly this is done for each and every character to finally produce “XIRCIA” as the cipher text.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Table 2: Vigenere Character set Table

The program used for the messaging application will use not just a 26x26 matrix as a message can include of multiple other characters including numeric characters, special characters as well as lower case

symbols. A new more secure 64x64 matrix is used to further increase the security and difficulty of decrypting the cipher text without a decryption key.

### C. Hybrid encryption algorithm design

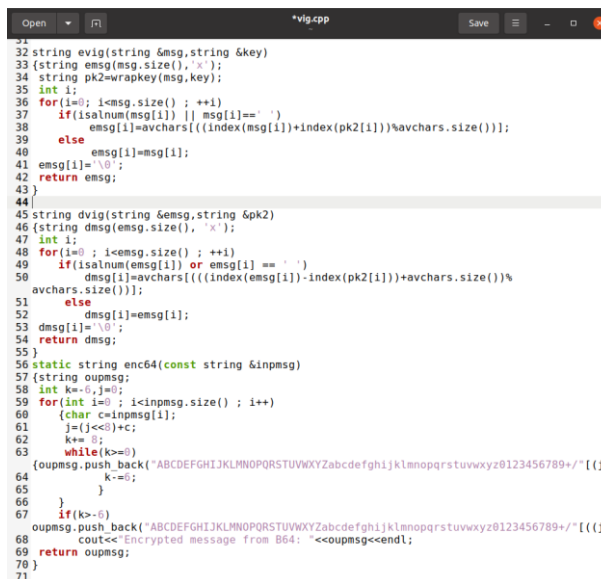
The hybrid encryption algorithm is designed through the use of Base-64 algorithm and the Vigenere encryption algorithm. The algorithm follows the order of Base-64 Algorithm followed by the Vigenere Algorithm, in order to maximize the efficiency as well as the security of the system.

An example for the Hybrid encryption algorithm, for the plain text “NETSEC” with encryption key “KEY”. The plain text is converted into B-64 which gives the first stage ciphertext as “TkVUU0VD”. The cipher text is then passed to the Vigenere algorithm where the first step carried out is the extension of the encryption key through the process of text wrapping where the key is repeated to match the message length. The new encryption key is now “KEYKEYKE”. The two string are then compared character by character in the Vigenere character table to get the output as “sOItNnU7”.

Wrap text process is also an essential part of improving the security of the encrypted code as it will prevent anyone who has the key to be unable to decrypt as they will not be able to identify the message or key length due to the extension in ciphertext message due to the B64 algorithm.

## 6. Code:

The following is a part of the functional algorithm implemented through the use of C++ programming language in the application.



```
32 string evig(string &msg,string &key)
33 {string emsg(msg.size(),'x');
34 string pk2=wrapkey(msg,key);
35 int i;
36 for(i=0; i<msg.size(); ++i)
37 if(!isalnum(msg[i]) || msg[i]==' ')
38 emsg[i]=avchars[((index(msg[i])+index(pk2[i]))%avchars.size())];
39 else
40 emsg[i]=msg[i];
41 emsg[i]='\0';
42 return emsg;
43 }
44
45 string dvig(string &msg,string &pk2)
46 {string dmsg(msg.size(),'x');
47 int i;
48 for(i=0; i<msg.size(); ++i)
49 if(!isalnum(msg[i]) || msg[i]==' ')
50 dmsg[i]=avchars[((index(msg[i])-index(pk2[i]))+avchars.size())%
avchars.size()]);
51 else
52 dmsg[i]=msg[i];
53 dmsg[i]='\0';
54 return dmsg;
55 }
56 static string enc64(const string &inmsg)
57 {string oupmsg;
58 int k=0,j=0;
59 for(int i=0; i<inmsg.size(); i++)
60 {char c=inmsg[i];
61 j=(j<<8)+c;
62 k++;
63 while(k==6)
64 {oupmsg.push_back("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"[(j
65 k-=6;
66 }
67 if(k>=6)
68 oupmsg.push_back("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"[(j
69 cout<<"Encrypted message from B64: "<<oupmsg<<endl;
70 return oupmsg;
71 }
```

Diagram 3: C++ code of Vigenere encryption

## 7. Result

The image below shows the implementation of the hybrid encryption algorithm for the message “Hello” and also shows the step by step result of each encryption step in the algorithm as well as the decryption process.

```
i5mail@i5mail-VirtualBox:~/Desktop/NS$ g++ vig.cpp -o vig
i5mail@i5mail-VirtualBox:~/Desktop/NS$ ./vig
Message to send: Hello
After Base 64 Encryption : SGVsbG8
New Extended Key : 12aB4i1
Hybrid Encrypted message: IxVT50Y
Extended Key For Decryption Process : 12aB4i1
After Vigenere Decryption : SGVsbG8
Decrypted message: Hello
i5mail@i5mail-VirtualBox:~/Desktop/NS$
```

Diagram 4: Hybrid encryption algorithm encryption and decryption output

In order to implement the real-life scenarios using this hybrid encryption algorithm, this method has been deployed into an server-client chat application made out of socket programming using Ubuntu as the main operating system. In this deployment the private key is randomly generated with a length that is randomly generated using built in functions. The server alone possesses the function to generate these private keys randomly which are then passed to the client upon creation of a socket connection between server and the client. These private are used to encrypt and decrypt the messages by both client and the receiver when sending and receiving messages.

```
riwaz@Riwaz-VirtualBox: ~
riwaz@Riwaz-VirtualBox:~$ g++ client.cpp -o client
riwaz@Riwaz-VirtualBox:~$ ./client 127.0.0.1 2204
NETWORK SECURITY PROJECT::WIN SEM 2020-21
To safely stop the chat type 'exit'!!!!
Passkey Successfully Received From The Server.....
To Server : This is implementation of Network Security Project
From Server : Done by

To Server : Seyed Ismail Mohamed(19BCE2509)
From Server : Riwaz Udas(19BCE2532)

To Server : Pranav Kumar Jha(19BCE2579)
From Server : Raghav Asawa(19BCE0861)

To Server : Thank you!!!
█
```

Diagram 5: Client-Side messaging application

```
riwaz@Riwaz-VirtualBox: ~
riwaz@Riwaz-VirtualBox:~$ ./server 2204
NETWORK SECURITY PROJECT::WIN SEM 2020-21
To safely stop the chat type 'exit'!!!!
Private Key Has Been Successfully Generated.....
Private Key Generated : iEjeh*****
Private Key Has Been Sent To The Client
From Client : This is implementation of Network Security Project

To CLIENT : Done by
From Client : Seyed Ismail Mohamed(19BCE2509)

To CLIENT : Riwaz Udas(19BCE2532)
From Client : Pranav Kumar Jha(19BCE2579)

To CLIENT : Raghav Asawa(19BCE0861)
From Client : Thank you!!!

To CLIENT :
```

Diagram 6: Server-Side messaging application

## 8. Conclusion

The algorithm has been successfully implemented into a messaging application through the use of socket programming, communication between two users is now performed securely after all messages are encrypted using the hybrid encryption system. This application can be further implemented into any system regardless of the purpose of the application. The application fulfills all requirements as discussed in the project. The encryption algorithm can be further strengthened depending on the scenario of the requirement by adding another commonly used encryption algorithm like the RSA, DES or AES encryption system after the Vigenere algorithm.

## 9. References

1. Yu, L., Wang, Z., & Wang, W. (2012). *The Application of Hybrid Encryption Algorithm in Software Security*. 2012 Fourth International Conference on Computational Intelligence and Communication Networks. doi:10.1109/cicn.2012.195
2. Dr. Kamaljit I. Lakhtaria “Protecting Computer Network with Encryption Technique: A Study” published in International Journal of u- and e- Service, Science and Technology Vol. 4, No. 2, June, 2011
3. Milon Biswas, Nazmus Sakeef, Hisham Siddique. Exploring Network Security Using Vigenere Multiplicative Cipher Encryption and Implementation, DOI: [10.21203/rs.3.rs-58356/v1](https://doi.org/10.21203/rs.3.rs-58356/v1)
4. <https://www.geeksforgeeks.org/vigenere-cipher/>
5. <https://levelup.gitconnected.com/what-is-base64-encoding-4b5ed1eb58a4>
6. <https://www.base64encoder.io/learn/>
7. Forouzan, B.A. (2007). *Cryptography and Network Security*. (Special Indian Edition). McGraw Hills Company Inc. New York