Oriental Journal of Computer Science & Technology Vol. **4**(1), 67-74 (2011)

Construction of High Performance Stream Ciphers for Text Web Browsers using Adders

RAJ KUMAR and V.K. SARASWAT

Institute of Computer & Information Science, Khandari,
Dr. B.R. Ambedkar University, Agra - 2 (India).

(Received: January 13, 2011; Accepted: February 18, 2011)

ABSTRACT

There are several security measures to protect the sensitive, confidential and secretive data over the internet. One of the basic mechanism is to apply a secret code of encryption. The basic problem with encryption has been speed, the reliability of mechanism and the compatibility with different web browsers. We have several encryption algorithms like RSA, IDEA, TEA and DES. These algorithms are developed by different people and written in many forms i.e. in different programming languages.

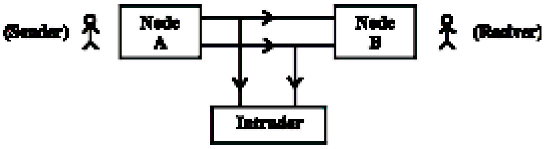
Here we are proposing a model of stream cipher based on encryption techniques with digital adder's operation. The text is translated into hex code using some secret key (say password). The secret key is only shared by sender and receivers. So only authorized personnel can access the data sent.

Our model proposes to use n-bit serial adder to encrypt data with fast speed and high reliability. The encrypted data are converted into hexadecimal to enhance the compatibility of the proposed mechanism with different machines and platforms. We have performed an analytical analysis to determine the best working algorithms for text web browser.

Key words: Cipher, browser, adder, encryption, reliability.

INTRODUCTION

The information stored is secured by several security measures. But information get the problem of susceptibility when one send it to some another computer on internet. As some unauthorized person like hacker can incept, conceal or stolen the fully or partial part of information.

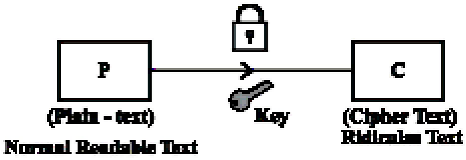


To come out of the problem, here sender makes the text ridiculous by using encryption methods. Thus even if hacker managed to intercept or steal the information send, it would be meaningless or futile for them.

The reliability of the mechanism is measured by the efforts made by the intruder and wasted time to decrypt or cease the original information.

Conceptual framework

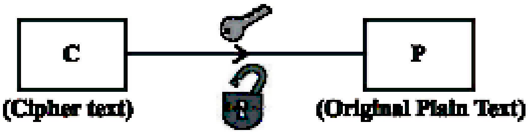
The this paper, we have proposed the model of stream ciphers fabricated from adders.



(Process of Encryption)

The basic idea of encryption is converting a normal text into unreadable text to others. This is referred as cipher text. The size of encrypted data depends on the length of bits in the secret key. The length of the key determine the probabilities which one ought to figure it out all its possible key values.

On the other side for the receiver to be able to read the cipher text, the text has to be decrypted again simply be reversing the process using the same key.



(The Process of decryption)

Encryption Technique

We are considering a single line of text to implement the encryption technique presented in this paper. The input plain text is shown as follows.

R A J K U M A R

Now, as the above text consists of eight characters we will store them as series of bytes that have been concerted into hex code.

R	A	J	K	U	M	A	R	Character
52	41	4A	4B	55	4D	41	52	Hexadecimal

Now we are using adder's operation to encrypt the text "RAJ KUMAR" using the secret key or password as "TAJ MAHAL". So the encryption key is below.

TAJMAHAL

Here again we will store these eight characters as hex code.

T	A	J	M	A	H	A	L
54	41	4A	4D	41	48	41	4C

So we will encrypt the text "RAJKUMAR" be applying adder function. We can consider a series of 8 – adders to encrypt the text taken above in order to enhance the performance of encryption.

Logic of Adder Function

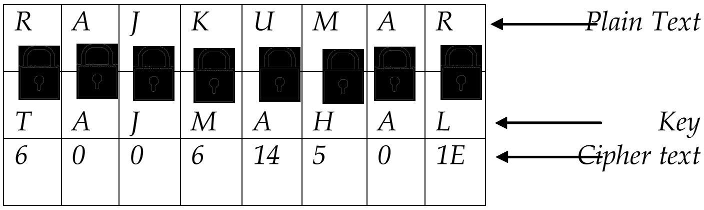
The output of adder function is zero if both (input data A) and key (input B) are identical, and the output of adder function is one if both (data input A) and key (input B) are not same. The output is represented by A + B.

Following is the truth table of adder function listed in figure 1.

Input A	Input B	Output A + B
0	0	0
0	1	1
1	0	1
1	1	0

(Figure 1 : Output of Adder function)

Now, we use the adder function to encrypt the text “RAJKUMAR” by using the key “TAJMAHAL” where T encrypts R, A encrypts A, J encrypts J, M encrypts K, A encrypts U and so on. At the end L encrypts. R. The entire process of encryption is shown below.



R	*A*	*J*	*K*	*U*	*M*	*A*	*R*
🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒
T	*A*	*J*	*M*	*A*	*H*	*A*	*L*
6	*0*	*0*	*6*	*14*	*5*	*0*	*1E*

Fig. 2: Result of Encryption

Output table of encryption process

The result of encryption in listed above in figure 2,

Following are the tables of their encryption process.

Table 1 : Encryption of R using key T

Char	*Hex code*	*Decimal*		*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*		*Decimal*	*Hex code*
R	*52*	*82*	=	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*0*			
T	*54*	*84*	=	*0*	*1*	*0*	*1*	*0*	*1*	*0*	*0*			
				0	*0*	*0*	*0*	*0*	*1*	*1*	*0*	=	*6*	*6*

Now we show the encryption of A with key A

Table 2 : Encryption of text A with key A

Char	*Hex code*	*Decimal*		*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*		*Decimal*	*Hex code*
A	*41*	*65*	=	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*1*			
A	*41*	*65*	=	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*1*			
				0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	=	*0*	*0*

Now we show the encryption of text J with key J.

Table 3 : Encryption of text J with key J

Char	*Hex code*	*Decimal*		*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*		*Decimal*	*Hex code*
J	*4A*	*74*	=	*0*	*1*	*0*	*0*	*1*	*0*	*1*	*0*			
J	*4A*	*74*	=	*0*	*1*	*0*	*0*	*1*	*0*	*1*	*0*			
				0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	=	*0*	*0*

Now we show the encryption of K with key M.

Table 4 : Encryption of K with key M

Char	Hex code	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code
K	4B	75	=	0	1	0	0	1	0	1	1			
M	4D	77	=	0	1	0	0	1	1	0	1			
				0	0	0	0	0	1	1	0	=	6	6

Now we show the encryption of text U with key A.

Table 5: Encryption of U with key A.

Char	Hex code	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code
U	55	85	=	0	1	0	1	0	1	0	1			
A	41	65	=	0	1	0	0	0	0	0	1			
				0	0	0	1	0	1	0	0	=	20	14

Now we show the encryption of text U with key A.

Table 6 : Encryption of M with key H

Char	Hex code	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code
M	4D	77	=	0	1	0	0	1	1	0	1			
H	48	72	=	0	1	0	0	1	0	0	0			
				0	0	0	0	0	1	0	1	=	5	5

Similarity we show the encryption of R with key L.

Table 7 : Encryption of text R with key L

Char	Hex code	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code
R	52	82	=	0	1	0	1	0	0	1	0			
L	4C	76	=	0	1	0	0	1	1	0	0			
				0	0	0	1	1	1	1	0	=	30	1E

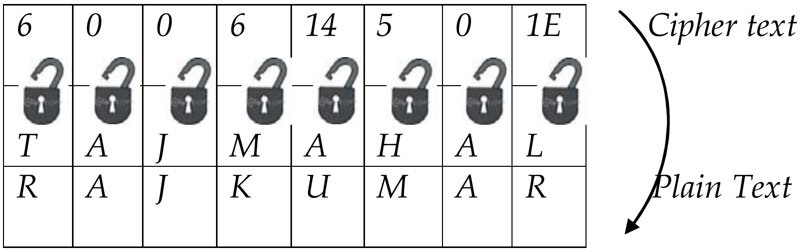
Thus our encrypted cipher text would be following (as shown in figure 2 also).

6	0	0	6	14	5	0	1E	←*Cipher text*

Now, at the receiver we get "RAJKUMAR" as "600614501E" which can be only decrypted by the authorized personnel having the key (or password) used here like "TAJMAHAL".

Decryption Technique

Conversion from the received cipher text to original plain text may be done by the reversal process of mechanism used in encryption. Once again the secret key (or password) will be applied in the reversal process. We have enumerated the output of the entire process.



6	0	0	6	14	5	0	1E
T	A	J	M	A	H	A	L
R	A	J	K	U	M	A	R

Fig. 3 : Result of decryption

Output tables of the decryption process

The result of decryption is shown in figure 3, following are the tables of their decryption processes.

Table 8 : Decryption of 6 with T

ode	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code
6	6	=	0	0	0	0	0	1	1	0			
(54)	84	=	0	1	0	1	0	1	0	0			
			0	1	0	1	0	0	1	0	=	82	52

Now, we show the decryption of 0 with key A

Table 9: Decryption of 0 with key A

Iex ode	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code	Char
0	0	=	0	0	0	0	0	0	0	0				
(41)	65	=	0	1	0	0	0	0	0	1				
			0	1	0	0	0	0	0	1	=	65	41	A

Similarly we find the decryption of 0 with J

Table 10 : Decryption of 0 with key J

ex ode	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code	Char
O	0	=	0	0	0	0	0	0	0	0				
(4A)	74	=	0	1	0	0	1	0	1	0				
			0	1	0	0	1	0	1	0	=	74	4A	J

Now we show the decryption of 6 with key M

Table 11 : Decryption of 6 with key M

ex ode	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code	Char
O	6	=	0	0	0	0	0	0	0	0				
(4D)	77	=	0	1	0	0	1	1	0	1				
			0	1	0	0	1	0	1	1	=	75	4B	K

Now we compute the decryption of 14 with key A.

Table 12 : Decryption of 14 with key A

ex ode	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code	Char
14	20	=	0	0	0	1	0	1	0	0				
(41)	65	=	0	1	0	0	0	0	0	1				
			0	1	0	1	0	1	0	1	=	85	55	U

Now we show decryption of 5 with key H.

Table 13 : Decryption of 5 with key H

ex ode	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code	Char
5	5	=	0	0	0	0	0	1	0	1				
(48)	72	=	0	1	0	0	1	0	0	0				
			0	1	0	0	1	1	0	1	=	77	4D	M

At the end we show the decryption of 1E with key L.

Table 14: Decryption of 1E with key L

Iex ode	Decimal		128	64	32	16	8	4	2	1		Decimal	Hex code	Char
1E	30	=	0	0	0	1	1	1	1	0				
(4C)	76	=	0	1	0	0	1	1	0	0				
			0	1	0	1	0	0	1	0	=	82	52	R

Thus we find the following text from the cipher text "600614501E" with secret key "TAJMAHAL".

R A J K U M A R

RESULTS

The proposed model of cipher stream shows that parallel bit adders can be used to encrypt the text faster. Further no. mathematical operation like modulo, division etc are needed and hence the performance of the proposed model would be high in comparison to other ciphers like IDEA, TEA, AES and RSA.

CONCLUSIONS

Analytically the cryptosystems are compared with their performance, reliability and efforts required to break them by some intruder, hacker or unauthorized personnel's. The proposed model of cryptosystem can use changeable secret key's (or password) to encrypt/decrypt the block of text. As the working of the proposed model is based on the simple logic of digital gates, it is very easy to implement them with any architectures. High performance with reliability is achievable by using adders of more no of bits.

REFERENCES

1. Huffman "A method for the construction of minimum redundancy codes Proc. IRE, **40**: 1098-1101 (1952).
2. Latha Pillai, "Huffman Coding" EXILINX, Virtex Series, XAPP616 (v1.0) (2003).
3. Whitefield Diffie, Martin E Hellman "New directions in cryptography" IEEE International Symposium on Information theory, Sweden, 21-24 (1976).
4. R.L. Rivest, A. Shamir, L. Adleman "A method for obtaining digital signatures and Public-Key Cryptosystems", Communications of the ACM **21**: 120-126 (1978).
5. Joffrey Hoffstein, Jill Pipher, Joseph H Silverman "NTRU – A Ring based public key cryptosystem" Lecture notes in Computer Science, Springer – Verlag, Berlin **1433**: 267-288 (1998).
6. Joffrey Hoffstein, Joseph H Silverman "Optimizations for NTRU" Proceedings of conference on Public Key Cryptography and Computational number theory, Warsaw, De Gruyter, (Sep 11 – 15), 77-88 (2000).
7. Collen Marie O'Rourke "Efficient NTRU Implementation" A thesis For Master of Science at Worcester Polytechnic Institute, (2002).
8. Karthik Thiagarajan "NTRU – A Public key ring based algorithm" A thesis for Master of Science, University of Taxas, Dallas (2003).

9. NTRU Cryptosystem, Technical Reports 2002 available at http://www.ntru.com

10. J.Hoffstein, J.Pipher, J.H. Silverman "NTRU : A new high speed public key cryptography system in Algorithmic number theory" (ANTS III), Portland, OR, June 1998, Lecture Notes in Computer Science 1423 (J.P. Buhler, ed.) Springer – Verlag, Berlin 267 – 288 (1998).