Volume 4, Issue 10, October 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Network Security Using Double Symmetric Key: Algorithm SKG 2.3

Satish Kumar Garg
Govt. P G College Ambala Cantt - 133001
Haryana, India

Abstract - In the present work, named algorithm SKG 2.3, the author has used double symmetric key for network security. For encryption and decryption of any text file, the symmetric keys are used at four stages : (1) shifting N1 leftmost characters to rightmost in a circular queue or vice versa (2) converting each of N character so obtained into binary form using 8-bit ASCII Code (3) swapping integral multiples of leftmost N2nd bits with corresponding rightmost bits till 4N bits if remainder of (8N+1)/N2 is zero, otherwise upto 8N bits and (4) finally converting 8N bits so obtained into text using 8-bit ASCII Code. This algorithm can be applied to any text consisting of 10 or more characters. The results obtained after application of this algorithm are very good.

Keywords: Encryption, Decryption, shifting characters to left or right, swapping of bits.

I. INTRODUCTION

The internet technology [1,2] is developing at very fast speed and is being used almost in every field. In this technology, computers are used to send and receive data. The confidential data may be bank statements, bank transaction, military information, confidential data of companies etc. There is always a possibility that any unauthorized person may intercept our data, so it is not safe to send confidential data from one computer to another computer. Hence the data should be protected from an unauthorized person otherwise any massive disaster may happen all-on-a- sudden. In order to make secure the system one should consider the security primary attributes such as confidentiality, integrity and availability, and secondary attributes such as authenticity, non-repudiation and accountability etc. There are a large number of methods and techniques to achieve these security goals, one of these is Cryptography. Cryptography[3,4] is the process used to make a meaningful message to appear meaningless. Cryptography is not the only means of providing information security, but rather one set of techniques. The cryptographic algorithm can be classified into two categories: (i) Symmetric Key Cryptography where one key is used for both encryption and decryption purpose. (ii) Public Key Cryptography where two different keys are used one for encryption and the other for decryption purpose. Due to massive computation the public key crypto system may not be suitable in security of data in sensor networks [5]. The author has already developed an algorithm named as algorithm SKG 2.0 which is successful for encrypting any text/string consisting of 10 or more characters [6]. In the present work, algorithm SKG 2.3, the author has integrated algorithm SKG 2.0 and a new variable that is shifting N1 leftmost characters to rightmost in a circular queue or vice versa. This algorithm can be applied to any text consisting of 10 or more characters. The results obtained after application of this algorithm are very good.

II. THEORY

The algorithm SKG 2.3 is based on the concept that each character is represented by a unique 8-bit code in ASCII Code system and if one or more bits are changed in a 8-bit code corresponding to any character, then corresponding character is entirely changed. When any text of 10 characters is converted into binary form we get 80 bits which contains about 50% of 0's and 1's each. Therefore, total number of possible combinations is about $80!/(40!)^2 = 1075\times10^{20}$. The Super Computer available is Teraflop which is capable of doing 10^{12} floating point calculations per second, so a teraflop super computer shall take about 3409 Years to find all possible combinations [6].

This is being done in the following steps :

(1) shifting N1 leftmost characters to rightmost in a circular queue or vice versa
(2) converting each of N character so obtained into binary form using 8-bit ASCII Code
(3) swapping integral multiples of leftmost N2nd bits with corresponding rightmost bits till 4N bits if remainder of (8N+1)/N2 is zero, otherwise upto 8N bits and
(4) finally converting 8N bits so obtained into text using 8-bit ASCII Code.

ENCRYPTION ALGORITHM (MENU DRIVEN GUI PROGRAM)
// Read the text input and check length of Input, if less than 10, give error message
Step 1: Start
Step 2: Read input text N

Step 3: If (N.length()< 10)
Print error message that program is not applicable;
// Shift leftmost N1 characters to rightmost of the string of characters and vice versa)
Step 4: initialize character array a1[l]// copying each character of text N to character array a1[l]
Step 5: Read value of N1 and Shifting
Step 6: If shifting rightmost characters to leftmost, then go to step 8
Step 7: If shifting leftmost characters to rightmost
for each character starting from (N2+1) to N
copyeach character bit to array a1
for each character starting from 1st character upto N2
copy(append) each character to a1
Step 8: for each character starting from ((N+1)- N2) to N
copy each character to array a1
for each character starting from 1st character to (N-N2) character
copy (append) each character to a1
//Convert the text of N characters to binary form using 8-bit ASCII Code
Step 9: initialize character array str[]//copying each character of text N to character array str[]
Step10: for(i=0; i<N.length(); i++)
{
If(i==N.length() – 1)
Str[i] = N.substring(i);
else
Str[i] = N.substring(I, i+1);
}
Step 11: initialize byte array bytes[] // copy each text character of character array converted to byte char
Step 12: for(i=0; i<str.length; i++)
{
bytes[i]= (byte) Str[i];
}
Step 13: for each byte in array, convert each byte to binary bits and create string of those binary bits
// Interchange the leftmost integral multiple of N2 bits with corresponding rightmost integral multiple of N2 bits
Step14: initialize l = length of binary string //this will be used at number of places in program
Step 15: initialize integer j (to store length to be traversed [loop through])
Step 16: Read value of N2
Step 17: integer r = remainder of (binary+1) modulus N2
Step 18: if (r==0)
j = l/2;
else
j = l;
Step 19 : for(i=1;N2*i<=j;i++)
{
ch = charAt(N2*i-1);
Replace/ set charAt((N2*i)–1)th position with charAt((l+1)-(N2*i))th position;
Replace/ set charAt((l+1)-(N2*i))th position with char stored in variable ‘ch’;
}
// Convert 8N bits so obtained into the text of N characters using 8-bit ASCII Code
Step 20 : String s1 = ‘’;
Step 21 : String s=binary; // to store the binary string
Step 22 : char nextChar; // declare the variable to store next significant character in string
Step 23 : for(inti = 0; i<s.length(); i += 8) //this is a little tricky, as we want [0, 7], [9, 16], etc
{
nextChar = Integer value of s.substring(i, i+8);
s1 = s1 + nextChar;
}
Step 24: return s1 to output file **// return the final string**

III. IMPLEMENTATION OF ALGORITHM SKG 2.3, RESULT AND DISCUSSION

The author has implemented the said algorithm SKG 2.3 on Java platform for different values of N1= 1 to (N-1), direction of shifting the characters to left/right and N2 =3 to 8N/3. e.g., for input text :
Located in Kurukshetra, the land of Bhagwadgita, Kurukshetra University is a premier institute of higher learning in India. Its foundation stone was laid on January 11, 1957 by Bharatratna Dr. Rajender Prasad, the first President of the Indian Republic. The output is given Table 1 :

Table 1: Comparison of Encrypted Output Text of Algorithms SKG 2.0 and SKG 2.3

S.No.	Comparison of Encrypted Output Text of Algorithms SKG 2.0 and SKG 2.3
1.	Encrypted Output Text of Algorithms SKG 2.0 for N1=3
	╬╞╢ &G« µj~ógrMóg$v≈$,vó$?ε$N. oεL╓bτ6,D4$ ├Ω Fª&wóD ├j4N"Mªd fóLfóL╧2@E╟TD∞,▌ ¼4¿ E═ L ├«t╟rv≥o╢ w L╬εµvdnε □÷oª$□ ¬f&760\g▒} ªfrMvΓt▀vL?ªEjWΓ$f╟ 760$n«,╓*L7▒ó╗ªLF DΩD╟<╓Ω\εjv¬?J,ε6N ╓«^ª╥uó,╓µ$ ├Ωd?6@Dbtv╟2µ6.D0F*4^εD∩n,▌ f?760$µ.╬≥"
	Encrypted Output Text of Algorithms SKG 2.3 for N1=2, Left Shift and N2=3
	≈2T╬6D∩.Kv╟▒6~╢ µ2, fdE*dτ&?ªN760,╟ng$& _.<"╟ó\K Lτ&vτrJ vG"╧v<?j>?jV b▀bΣ¿ ╟ 4═¼╓n∩vZ f■$6?ª$EΩ∩$&6≥.?760f÷▒?"dµ≥dDε< ┌$d ├▒&~▒v╢ τrwjε2FóεEftEó<ª*.Ωt?$^ª▒⁴ªJ Eó ├▒╓.?N6.▒vπ N.$VΩTµj>╚$$?.τ"∩ΓVbf÷ov ~$$ .tóL&? ├≥,N¬RErD&$g?F
	Encrypted Output Text of Algorithms SKG 2.3 for N1=2, Right Shift and N2=3
	□b,OóHDv?"f?7604V.g≥fr$.▒εnL*NF▒fDón &góL╟jEn$ov6WóZLt\b w. F.GóK G lñ┐ -E4 ┼ Oª>vóPEvd&?2╟ólDód≈*^Lrf╬*nr,■f╟.760Mt?&f╚$d╬$d□╢dGª&~$Lτ2v?6o÷∩.,n▒ ╟rMj4?╢$*? N▀ n▒^Gª~╬v*EóLfª ├≥.G«@ E0f▒dgól«óJ$d÷ 6wª$Dóg $╟n<ª6N?«Lτ≥~╢gª, ├µtsΓ
2.	Encrypted Output Text of Algorithms SKG 2.0 for N1=5
	F?wD|/& Lf"evT!#|dTrs,$t"┐╟`n&$nNjlDf7#df`6c¼ J5bσnr"wΣR@`GjLVg" τL|3bφV760!2dVl-9σV ),g ╘H6't─760%v┤L`'*uV&ge÷fc.τ`f2M╬dc#╛ $@ ~#▒Ff7>ddT!□■ rv?~D76073gd#k⌠FfrNΣn?#÷|s3, 037ºJ1"F╟@03d╓H~. e?d:,ñVh"7ε─lz"╨V`3#⌠¼ 6jσ?Fc g⌠760rσv@&7nT 'f$⌠@or═Nlc!~?R'"e ├Dc3
	Encrypted Output Text of Algorithms SKG 2.3 for N1= 2, Left Shift and N2=5
	c3Σ─db+nñJ5rσ╬rb7d÷@. d∞D*~σnDj/f760B*cg╓`$'}t@$"═ ╘R□+gLd6rσTn) vDr1+t|760k#ñD7602be╟@%`ñ╟N14φ ╘T>gñεf"zmf`%"┐ ╟d!bε∞F'r²n~⌠╟h&"]⌠rb$∩tF$qΣ╟ff2gtF$g4╓`{"∞d@n"□ ε760!n⌠@2+ñ4.b5?□'÷ⁿ J*e╓@~2e ╘n#bD÷."RßND n&σVR"e÷@$<ñt@g"µ∞r36 Trg!mdD&$ñNn 4∞D(>Σ╟hn"╥Dp7 l∞jo
	Encrypted Output Text of Algorithms SKG 2.3 for N1= 2, Right Shift and N2=5
	#c╟Nbk$u─760#~▒Nt2'ov`%&vΣ$"f∞D$#εD760g$ñFh!5g─L'k`D " u÷T#cld\2!ñtn#fe ╓ Z#6iHs σñX8%}╟L02m╬zv+pTtg o ├760")τ╟l0"|Σhz,}εF*+n$hffm─." t╓760&□±Nd#$φn FbcΣεf%bτDr">eld /nñj#.σdR1bÑ?$"c-b²J*3µDtz3tΩ@"Ff«760ZsεDnf%v?x2cg─lf"tHD bvmvr62D÷D;)t─N>"□├760>jσ?@f6mdf*σ⌠tb>
3.	Encrypted Output Text of Algorithms SKG 2.0 for N1=5
	F?wD|/&Lf"evT!#|dTrs,$t"┐╟`n&$nNjlDf7#df`6c¼ J5bσnr"wΣR@`GjLVg" τL|3bφ V760!2dVl-9σV ),g ╘H6't─760%v┤L`'*uV&ge÷fc.τ`f2M╬dc#╛ $@ ~#▒Ff7>ddT!□■ rv?~D76073gd#k⌠FfrNΣn?#÷|s3, 037ºJ1"F╟@03d╓H~.e?d:,ñVh"7ε─lz"╨V`3#⌠¼ 6jσ?Fc g⌠760rσv@&7nT 'f$⌠@or═Nlc!~?R'"e ├Dc3
	Encrypted Output Text of Algorithms SKG 2.3 for N1= 4, Left Shift and N2=5
	6g⌠ @&2O⌠Z7kglD6"u?7606:edc,d$f&2 ├∞H/%e─N#fe¼760 %vtB{*ePRk"En `v'bV`~+$╟r*#$tRo/}@R*kn÷|)$etL(-÷@+w∞@Rb~edR&;n ├760k,ñlf.ke. &τ$f/wnDH>;on7603&□╨Db7e÷ &sm─ ',$ε@f7uVpb!Ñ,(;;%? ")$µ@#"eTR#$ε─ "╛ ?Z#j⌡nl%"$tr#c⌡Σ*vld760.;µvTjBµDR!f⌡Ntb/vt'"$lFn+σNR'd⌠B${≈Lf##
	Encrypted Output Text of Algorithms SKG 2.3 for N1= 4, Right Shift and N2=5
	*.φBDg#⌡td& φN(g÷Tjs*e ╘Z#n ThobⁿdF&2oF *σ ├^+$g∞tc.4╬\0uor@'vf─.m╓Dr!²TPh)≈?@ µDLk'µ$@f!t∞T76σ$Fn"∞Lfbwf?Dgs÷n@&wñ∞f Kn ╘@a.┤lt3 v╬Tfv⌡tH'.4V\'>σ?V!c┤L`+&$╬njZ⌡N\#²ñ;,ñ7%┤fPjB∞dRc&f─\.#4─R.b ├╘B%.d─r"µΣR#f,$t '4├`0#t$P0'gld'n⌠ñN.rdlD εd@c~0╓D2g

From Table 1, it is clear that if we change even a single variable (N1 or direction of shifting the characters either Left or Right or/and N2) then output of the Algorithm SKG 2.3 is entirely different. The algorithm SKG 2.3 is successful for encrypting any text/string consisting of 10 or more characters. Minimum time required to decryt any text/string consisting of 10 or more characters is about 3409 Years to find all possible combinations, which is sufficiently large to decrypt any text.

IV. CONCLUSION

The proposed scheme, algorithm SKG 2.3, was tested in Java platform for different values of N1 = 1 to (N-1), left/right shifting and N2= 3 to 8N/3. In all cases the result came as per expectation. It has been estimated that to crack the code we will require more time than the data will reside on the medium to travel. So, it can be said that the proposed scheme will produce an efficient secured algorithm for data transfer in both wired and wireless networks.

REFERENCES

[1] Satish Kumar Garg, Review of Secured Routing for Wireless Ad hoc Network, *International Journal of Computing and Business Research*, Vol. 2 (1) 2011.

[2] Satish Kumar Garg, Wireless Network Security Threats, *International Journal of Information Dissemination and Technology*, Vol. 1 (2) 2011.

[3] T. Karygiannis and L. Owens, *Wireless Network Security*, NIST Special Publication, 2002

[4] William Stallings, *Cryptography and Network Security: Principles and Practice*, Prentice Hall, 5th Edition,2011.

[5] R. H. Karpinski, Reply to Hoffman and Shaw, *Datamation*, Vol. 16(10) p. 11 (Oct. 1970)

[6] Satish Kumar Garg, Cryptography By Swapping Bits : Algorithm SKG 2.0, *International Journal of Information Technology and Knowledge Management*, Vol.7 (2), 14-16, 2014

