International Journal of Emerging Engineering Research and Technology
Volume 3, Issue 11, November 2015, PP 1-6
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Design and Implementation of Hybrid Rc4 and Sha-2 Algorithms for Wifi Applications

M. Niharika[1], Ravindharan Ethiraj[2]

[1]*Department of ECE, Stanley College of Engineering and Technology for Women, Hyderabad, India (PG Scholar)*
[2]*Department of ECE, Stanley College of Engineering and Technology for Women, Hyderabad, India (Associate Professor)*

ABSTRACT

In this paper we are designed a stream cipher and hash algorithm. Stream cipher is rc4 algorithm which stands for Rivest Cipher 4 and hash algorithm is md5 which stands for message digest 5. RC4 has been used in various protocols like WEP and WPA (both security protocols for wireless networks) as well as in TLS.md5 for authentication these two algorithms based on generated a key 40 bit .a generated key and plaintext combined with xor operation then generated cipher text .cipher text and generated key combine with xor operation then generated plaintext .the proposed algorithm is rc4 and sha2 algorithm is designed and synthesized by using Xilinx 13.2 tool.

Keywords: stream cipher, hash algorithm.

INTRODUCTION

A stream cipher is an encryption algorithm that encrypts 1 bit or byte of plaintext at a time. It uses an infinite stream of pseudorandom bits as the key. For a stream cipher implementation to remain secure its pseudorandom generator should be unpredictable and the key should never be reused. Stream ciphers are designed to approximate an idealized cipher, known as the One-Time Pad. The One-Time Pad, which is supposed to employ a purely random key, can potentially achieve "perfect secrecy". That is, it's supposed to be fully immune to brute force attacks. The problem with the one-time pad is that, in order to create such a cipher, its key should be as long as or even longer than the plaintext. In other words, if you have 500 Megabyte video file that you would like to encrypt, you would need a key that's at least 4 Gigabits long. Clearly, while Top Secret information or matters of national security may warrant the use of a one-time pad, such a cipher would just be too impractical for day-to-day public use. The key of a stream cipher is no longer as long as the original message. Hence, it can no longer guarantee "perfect secrecy". However, it can still achieve a strong level of security Stream ciphers encrypt bits individually. This is achieved by adding a bit from a key stream to a plaintext bit. There are synchronous stream ciphers where the key stream depends only on the key, and asynchronous ones where the key stream also depends on the cipher text. The stream cipher is an asynchronous one. Most practical stream ciphers are synchronous ones an example of an asynchronous stream cipher is the cipher feedback (CFB) mode

RC4, which stands for Rivest Cipher 4, is the most widely used of all stream ciphers. It's also known as ARCFOUR or ARC4. RC4 has been used in various protocols like WEP and WPA (both security protocols for wireless networks) as well as in TLS. Unfortunately, recent studies have revealed vulnerabilities in RC4, prompting Mozilla and Microsoft to recommend that it be disabled where

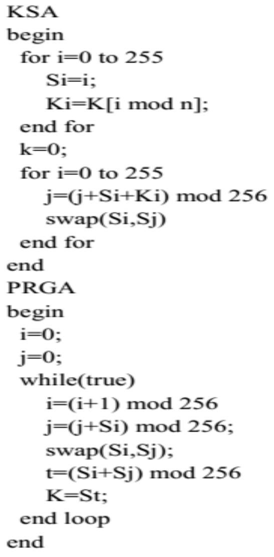
Address for correspondence:

niharika1195@gmail.com

possible. In fact, RFC 7465 prohibits the use of RC4 in all versions of TLS.These recent findings will surely allow other stream ciphers (e.g. SALSA, SOSEMANUK, PANAMA, and many others, which already exist but never gained the same popularity as RC4) to emerge and possibly take its place.

RC4 ALGORITHM

In cryptography, RC4 (also known as ARC4 or ARCFOUR meaning Alleged RC4, see below) is the most widely-used software stream cipher and is used in popular protocols such as Secure Sockets Layer (SSL) (to protect Internet traffic) and WEP (to secure wireless networks). While remarkable for its simplicity and speed in software, RC4 has weaknesses that argue against its use in new systems. It is especially vulnerable when the beginning of the output key stream is not discarded, nonrandom or related keys are used, or a single key stream is used twice; some ways of using RC4 can lead to very insecure cryptosystems such as WEP.RC4 generates a pseudorandom stream of bits (a keystream) which, for encryption, is combined with the plaintext using bit-wise exclusive-or; decryption is performed the same way (since exclusive-or is a symmetric operation). (This is similar to the Vernam cipher except that generated pseudorandom bits, rather than a prepared stream, are used.)



```
KSA
begin
  for i=0 to 255
      Si=i;
      Ki=K[i mod n];
  end for
  k=0;
  for i=0 to 255
      j=(j+Si+Ki) mod 256
      swap(Si,Sj)
  end for
end
PRGA
begin
  i=0;
  j=0;
  while(true)
      i=(i+1) mod 256
      j=(j+Si) mod 256;
      swap(Si,Sj);
      t=(Si+Sj) mod 256
      K=St;
  end loop
end
```

MD5 HASH ALGORITHM

Takes as input a message of arbitrary length and produces as output a 128 bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest. The message is padded so that its length is congruent to 448, modulo 512.Means extended to just 64 bits shy of being of 512 bits long. A single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits equals 448 modulo 512. A 64 bit representation of b is appended to the result of the previous step. the resulting message has a length that is an exact multiple of 512 bits. A four-word buffer (A, B, C, D) is used to compute the message digest.– Here each of A, B, C, D, is a 32 bit register. These registers are initialized to the following values in hexadecimal A: 01234567, B: 89abcdef, C: fedcba98, D: 76543210.Process message in 16-word blocks and four auxiliary functions that take as input three 32-bitwords and produce as output one 32-bit word(X, Y, Z) = XY v not(X) Z,G(X, Y, Z) = XZ v Y not (Z),H(X, Y, Z) = X xor Y xor Z,I(X, Y, Z) = Y xor (X v not (Z)) if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z), G(X,Y,Z),H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased. The message digest produced as output is A, B, C, D. That is, output begins with the low-order byte Of A, and end with the high-order byte of D.

Block Diagram

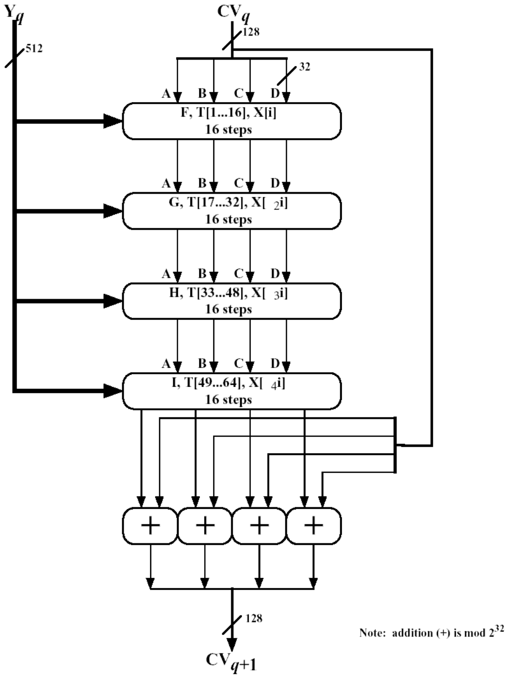
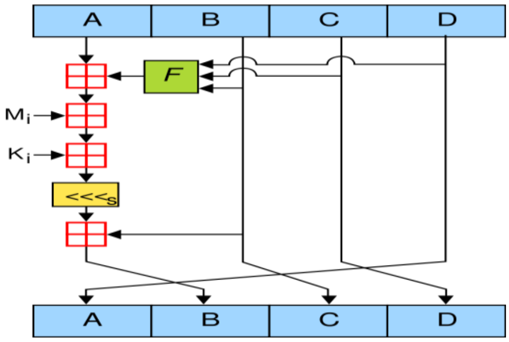
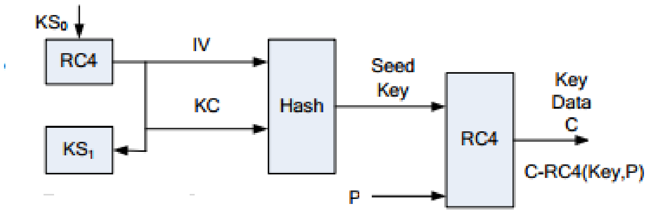


Figure 9.2 MD5 Processing of a Single 512-bit Block (MD5 Compression Function)

Step Logic for MD5

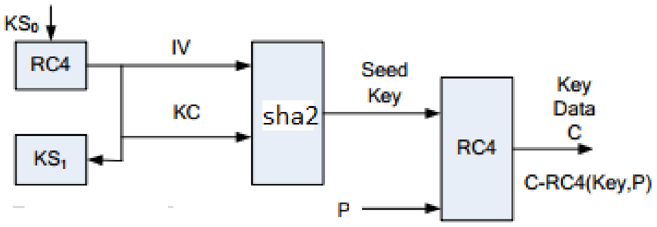


PROPOSED ALGORITHM



A uses the RC4 algorithm and KS0 to get a pseudo-random output stream, after moving the first 48 bytes away from the output stream, select the interception between 49 bytes and 52 bytes as IV, select interception between 53 bytes and 80 bytes as the user shared key KC, and select the interception between 81 bytes and 96 bytes as the next key KS1. A uses a certain combination of Hash functions to get the seed key, and then we finish the seed key generation stage. Finally, we get a new pseudo-random data stream with the seed key generated by RC4 algorithm; discard the first 48 bytes of the data stream and then encrypt the P. A. Send the encrypted result cipher text C(RC4(key,P)) to B. B decrypts the received data in the same way and gets the message data P

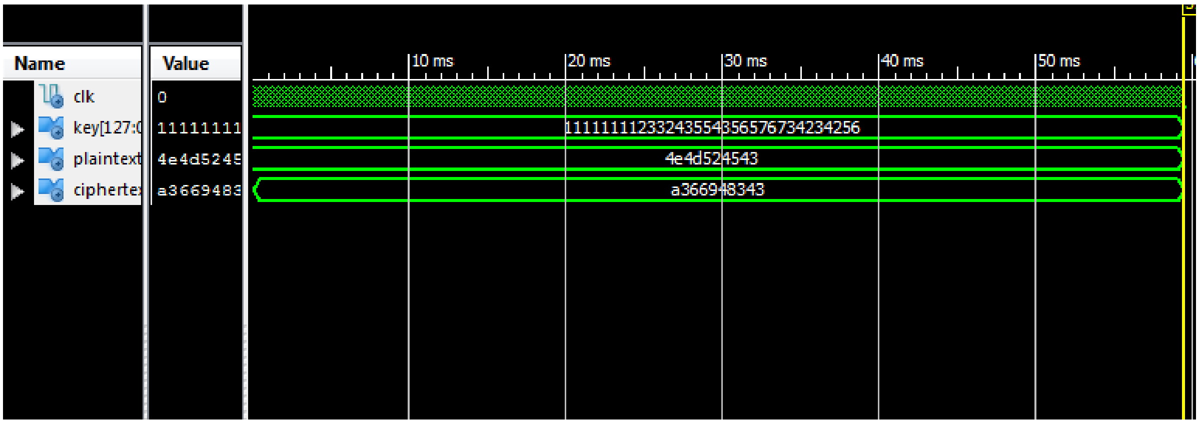
EXTENSION



A uses the RC4 algorithm and KS0 to get a pseudo-random output stream, after moving the first 48 bytes away from the output stream, select the interception between 49 bytes and 52 bytes as IV, select interception between 53 bytes and 80 bytes as the user shared key KC, and select the interception between 81 bytes and 96 bytes as the next key KS1. A uses a certain combination of sha2algorithm to get the seed key, and then we finish the seed key generation stage. Finally, we get a new pseudo-random data stream with the seed key generated by RC4 algorithm; discard the first 48 bytes of the data stream and then encrypt the P. A. Send the encrypted result cipher text C(RC4(key,P)) to B. B decrypts the received data in the same way and gets the message data P

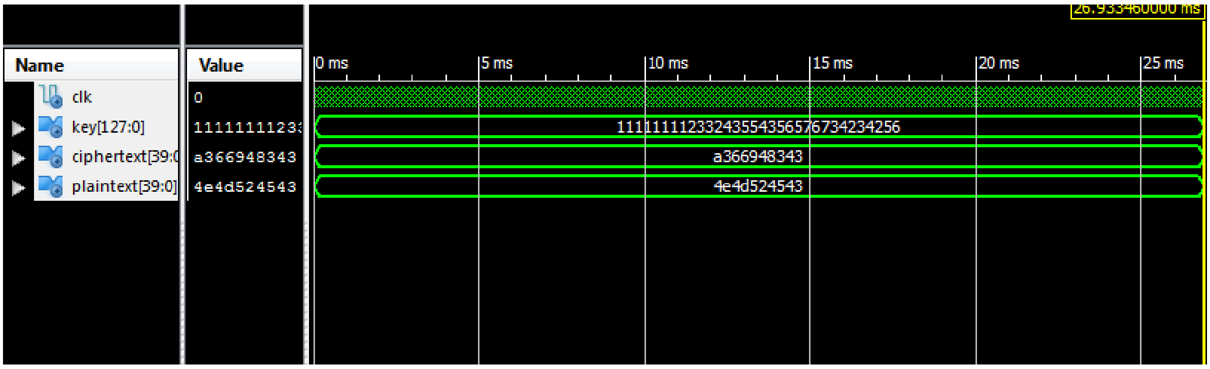
SIMULATION WAVEFORMS

Encryption



Name	Value
clk	0
key[127:(	11111111
plaintext	4e4d5245
ciphertex	a3669483

Decryption



Name	Value
clk	0
key[127:0]	111111112333
ciphertext[39:(	a366948343
plaintext[39:0]	4e4d524543

CONCLUSION

The hybrid rc4 and md5 algorithms discussed in this paper. *these two algorithms based on generated a key 40 bit .a generated key and plaintext combined with xor operation then generated cipher text .cipher text and generated key combine with xor operation then generated plaintext .the proposed algorithm is rc4 and sha2 algorithm is designed and synthesized by using Xilinx 13.2 tool.*

REFERENCES

[1] BORSE M, SHINDE H, Wireless Security & Privacy [J]. Personal Wireless Communications, ICPWC 2005.

[2] Yukio Mitsuyama, Motoki Kimura, Takao Onoye, Isao Shirakawa, Architecture of IEEE802.11i Cipher Algorithm for Embedded Systems. IEICE Trans. on Fundamentals, vol. E88-A, no.4, 2005, pp. 899-905.

[3] Jun-Dian Lee, Chih-Peng Fan, Efficient low-latency RC4 architecture designs for IEEE 802.11i WEP/TKIP. Intelligent Signal Processing and Communication Systems, Nov. 28, 2007, pp.:56-59.

[4] Maocai Wang, Guangming Dai, Hanping Hu, Security Analysis for IEEE802.11. Wireless Communications, Networking and Mobile Computing, 2008.

[5] Jovan Dj.Golic, Linear statistical weakness of alleged RC4 keystream generator. Advances in Cryptography EUROCRYPT'97, Lecture Notes in Compur, vol.1233, Springer, Berlin, 1997, pp.226-238.

[6] Scott R.Flunhrer and David A.McGrew, Statistical analysis of the alleged RC4 keystream genenrator, Fast Software Encryption 2000, Lecture Notes in Comput.Sci., Vol.1978, Springer Berlin, 2000, pp.19-30.

[7] Itsik Mantin and Adi Shamir, A practical attack on broadcast RC4, Fast Software Encryption 2001, Lecture Notes in Comput.Sci., vol.2355, Springer, Berlin, 2001, pp.152-164.

[8] Andrew Roos, A class of weak keys in the RC4 stream cipher, Post in sci.srypt, 2004.

[9] Grosul and D.Wallach, A related key cryptanalysis of RC4, Tech.Report TR-00-358, Department of Computer Science, Rice University, 2000.

[10] Mantin, Analysis of the stream cipher RC4, Master's thesis, The Weizmann Institute of Science, 2001.

[11] Ilya Mironov, (Not so) random shuffles of RC4, Advances in Crytography CRYPTO 2002, Lecture Notes in Comput.Sci., vol.2442, Springer, Berlin, 2002, pp.304-319.

[12] Scott Fluhrer, Itsik Mantin and Adi Shamir, Weakness in the scheduling algorithm of RC4, Selected Areas in Cryptography, Lecture Notes in Compute.Sci., vol.2259, Springer, Berlin, 2001, pp.1-24.

[13] Chuan-Chin Pu, Wan-Young Chung, Group Key Update Method for RC4 Stream Cipher in Wireless Sensor Networks, international conference on convergence information technology, 2007

AUTHOR'S BIOGRAPHY

M. Niharika, is pursuing her B.E 4th year in Electronics and Communication Engineering at Stanley College of Engineering and Technology for Women, O.U Affiliated College. Her research interest is VLSI Technology and Design and Communication Systems.

Prof. Ravindharan Ethiraj, has 50 years of teaching experience covering B.Sc., M.Sc., B. Tech. and M.Tech. At New Science College, he was actively involved in the development of the M.Sc. Applied Electronics Course during which time he pursued his Ph.D. Degree in Piezo Optic Studies in Mixed Cubic Crystals. During his tenure at Physics Department Osmania University, he was the Chairman Board of Studies in Electronics. He was also the Chairman, BOS in Hearing Language and Speech. He was associated with the development of Specializations of Microwave Physics at Koti Women's College, Fiber Optics at AV College and Communication and Cisco Networking at Nizam College. He was the Visiting Professor at Bosch and Lomb School of Optometry, Currently he is the Director, Research and Development at Stanley College of Engineering and Technology, where he is developing special courses of studies which aims at highlighting the importance of Mathematics and Pure Sciences in Engineering thereby bringing out the explorative and creative talents in engineering students.